Spring 5-1-2020

Computational Analysis of a new Planar Mixing Layer Flame Configuration to study Soot Inception

Carmen Ciardiello
cciardiello1@gmail.com

Computational Analysis of a new Planar Mixing Layer Flame Configuration to study Soot Inception

Carmen David Ciardiello

B.S., University of Connecticut, 2020

A thesis submitted in partial fulfillment of the requirements for graduating as an Honors Scholar at the University of Connecticut

2020

i. Table of Contents

Contents

ii. List of Figures

1. Abstract

The production of soot is omnipresent in society today. Soot is the product of many of the combustion processes that provide the bulk of the usable energy throughout the world. Furthermore, soot particulate poses a great danger to both the environment and all forms of life on Earth. It has proven to pollute ecosystems, foster health problems for human beings, and degrade air quality [1].

These dangers make studying and understanding soot particulate paramount for improving the quality of life. Thus, this study introduces a new flame configuration for studying soot inception. Presently, various common flame configurations have been found to be difficult to use to observe the change of gas to soot particulate in diffusion flame conditions [2]. This is attributed to the thickness of the mixing layer between fuel and oxidizing jets. This new configuration, the Planar Mixing Layer Flame, results in a thicker mixing layer that enables more robust probing capability.

This study is purely analytical in nature. No physical testing of the flame configuration was conducted. However, the ANSYS Fluent simulations and corresponding results laid out provide the basis for future modelling and the feasibility of the physical testing to generate the desired results.

2. Introduction

Soot emission affects combustion technologies that are responsible for more than 80% of the total energy production worldwide, with the dominance of combustion unlikely to subside in the foreseeable future [3]. Particulate matter derived from combustion, soot particulates, have proven to be detrimental to both human health and the acceleration of climate change. Exposure to soot particulates have be proven to foster health conditions such as asthma and exacerbate existing conditions such as heart and lung disease [4]. Soot emissions also contribute to the damage of crops and other vegetation, in addition to producing acid rain which endangers soil, lakes, and streams. This leads to the contamination of the human food chain. With regards to climate change, these emissions wreak havoc on air and water quality, weather patterns, the agricultural industry, and various ecosystems around the planet [4].

Particulates from diesel exhaust are comprised of approximately 80-90% carbon either as elemental carbon or organic carbon [5]. Elemental carbon is characterized as non-volatile and has the propensity to display strong light absorbing tendencies. Elemental carbon provides the core of particulate emitted from diesel engines. It is emitted over the course of a combustion process in the

form of soot particulates [5]. Organic carbon is different from elemental carbon in that it does not have the same non-volatile and light absorbing properties of elemental carbon. Organic carbon is associated with traffic, industrial applications, and the decay of carbon-based materials. Both of these forms of carbon lead to the toxicity of diesel particulate matter and in turn prove potent in visibility impairment and climate change [5].

A study done between 2002 and 2004 and written up in 2006 proved the fault of soot particulate in the decrease of visibility in two Korean cities [6]. In the study, the authors defined visibility impairment as a basic form of air pollutions able to be discerned by the human eye (i.e. able to be detected without special instrumentation) [6]. The authors attributed the volume of diesel powered automobiles present in major cities around the world. They found that elemental carbon was the largest contributor to visibility concerns in Seoul, the largest city in the country and one of the largest in the world [6].

Miller et all studied the link between air pollution and cardiovascular events in women by studying 65,893 women without any previous cardiovascular disease in 36 different metropolitan areas in the United States between 1994 and 1998 [7]. The authors, after adjusting for age, race and ethnicity, smoking status, and other

factors estimated hazard rations for the women's first cardiovascular event [7]. 1816 women were shown to have at least one fatal or nonfatal cardiovascular event. Examples included death from coronary heart disease and cerebrovascular disease, coronary revascularization, myocardial infarction, and stroke. They also found that for every 10 $\mu g/m^3$ increase in exposure to particulate matter, there was a corresponding 24% increase in risk of a cardiovascular event and a 76% increase in risk of death from a cardiovascular event.

Based on the studies presented above and many other in the academic universe, soot particulate has wide ranging implications on both humans, animals, and their environment. Soot particulate stands as a substantial risk to the continued long-term health of any of the aforementioned organisms or phenomena, thus studying such matter continues to be of great importance.

2.1 Background on Soot Particulate

Soot is a non-specific term that describes an amorphous substance made up carbon and hydrogen [8]. As alluded to in the discussion of previous studies, urban areas prove to have high concentrations of soot. This is yielded from the combustion of fuel and the properties of soot do not differ based on the fuel type or the impetus for the formation of the particulate [9]. The appearance of smoke can be attributed to the existence of soot. Soot transforms from a gaseous state to

solid in combustion process amongst fuel-intensive areas at relatively high temperatures [5], [10]. Based on the given operating conditions, hydrocarbons have shown the tendency to be absorbed by soot [11]. Newly incepted soot has a relatively high hydrogen factor, but as the particulate matures, that fraction begins to shrink as the carbon fraction grows. Specifically with diesel engines, one of the environments with a notably high concentration of soot, as the soot matures, one might observe small amounts of zinc, phosphorous, calcium, and other metals [5], [10]. Soot has no specific structure and its maturation is based on many chemical reactions and physical transformations. The lack of fundamental processes in soot inception and soot maturation make both difficult phenomena to model

2.1.1 Soot Formation

As explained above, there is no definitive articulation of how soot is developed. No researcher has been able to identify a consistent mechanism or patter. While many arguments regarding definitive processes in the study of soot remain controversial, there have been a handful of ideas that seem to be consistent in all processes containing soot [12], which has been summarized as the following: “soot begins with some precursors, the nucleation of heavy molecules occurs to form particles, surface growth of a particle proceeds by absorption of gas phase molecules, coagulation happens via reactive particle-particle collisions, oxidation

of the molecules and soot particles reduces soot formation" [13]. The precursors are specific agreed upon molecules, such as acetylene. The nucleation consists of transitioning the hydrocarbons from the gaseous phase to solid.

The mass of the particles increases during the combustion process as more gaseous hydrocarbon particles transition to the solid state as reactions are undergone between the solid soot and large hydrocarbons generated by pyrolysis of the flame products floating about after the oxidation of part of the fuel. The solid particles bond together in the coagulation step effectively simultaneously with the nucleation step, which can be interpreted as the first presence of soot particulate. The mass of the soot can be subsequently reduced if the enlarged particles are exposed to an oxidizing environment that turns them back into gaseous molecules. The rate of oxidization is based on the surface area of the soot particulate (more surface area, higher oxidization rates and vice versa). This is how the soot gets into the air and creates visibility impairment and hazardous breathing conditions.

2.2 Previous Modeling Techniques

Given the risks associated with soot emission, the modelling of soot and its inception has become paramount in efforts to mitigate the particulate. While this report proposes the introduction of a new flame configuration to understand soot formation, substantial efforts have been made in the past to take on the same task.

A paper by Omidvarborna et all from 2015 analyzed the existing soot models for diesel combustion. The authors opined that due to mandates by many governments across the globe, one of the top priorities of combustion researchers has been to formulate emission reduction techniques. Modeling soot proves to be an important factor in understanding soot formation and the effectiveness of emission reduction techniques [13] . The authors come to the conclusion that there exists a gap between the soot models and the actual inception of soot. They cite better comprehension of oxidation mechanisms and more realistic assumptions as avenues for model improvement. The difficulties associated with modeling mechanisms of these types is the complexity in the number of species in fuel, the nature of the combustion mechanisms, and the non-uniform interactions between species when simulating the mechanisms.

Generally models are categorized in three different manners: empirical models (which are calibrated with experimental data), semi-empirical models (combination of experimental data and theoretical equations), and theoretical mechanisms (based on chemical theory) [14]. The empirical models extrapolate experimental data to forecast soot particulate inception [15], [16], [17]. Empirical models can reaffirm our established understanding of soot inception, but since they are based off of previous experimental data, they do not allow for new

conclusions. The semi-empirical models mix theoretical models and previously recorded experimental data in such a way to simulate more trivial mechanisms [16]. Given the scope of this investigation, this classification of model will not provide a robust enough interpretation of physically realizable systems.

The last model type, theoretical mechanisms, contain hundreds of species and facilitate hundreds of reactions. These are most useful when trying to simulate and predict soot formation on any timeline, either past experiments or reactions with no established expectation of soot inception time or quantity. Due to the complexity of these models, computational power is a concern. These simulations can take hours or days to run.

They require residual convergence for any sort of meaningful analysis. The residuals of a simulation are the measure of the number of changes for a selected variable during one iteration of the solution. These residuals are scaled in such a way that they are normalized to one. For the solution to converge and be accepted for analysis, the residual for parameters of import must reach a specified value. Once this value is met, the solution achieves convergence if the residual for the parameter remains steady on a macro-level. One should expect small deviations around the specified value, but the value should not largely from then on out. The

convergence criteria is set at residuals of a few orders of magnitude, generally three or four.

2.3 Goals of the Present Study

Presently, there is not much known about soot inception in diffusion flames. Diffusion flames are the result of fuel species and oxidizer reacting with diffusion in a controlled region. These flames are inherently difficult to probe for the change of gas to soot particulate. In common configurations such as counterflow and axisymmetric coflow flames, it is impossible to totally separate phenomena to isolate soot inception. These phenomena include oxidation, geometrical constraints, sharp gradients, and nontrivial fluid dynamic couplings between the flame and probe. This study looks to develop a new way of facilitating fuel and oxidizer reactions. The proposed configuration is a coflow Planar Mixing Layer Flame (PML). The configuration is displayed in Fig.1 and Fig.2:

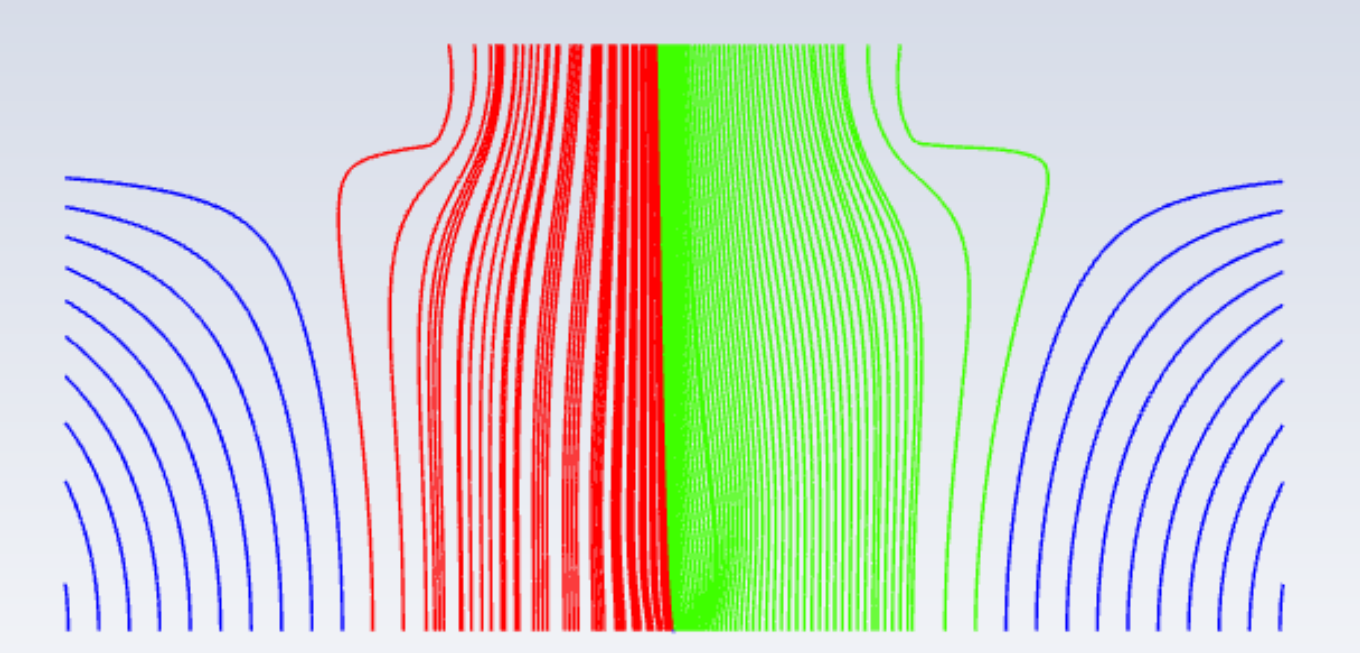

Figure 1: Streamline plot with colors corresponding to different inlet types

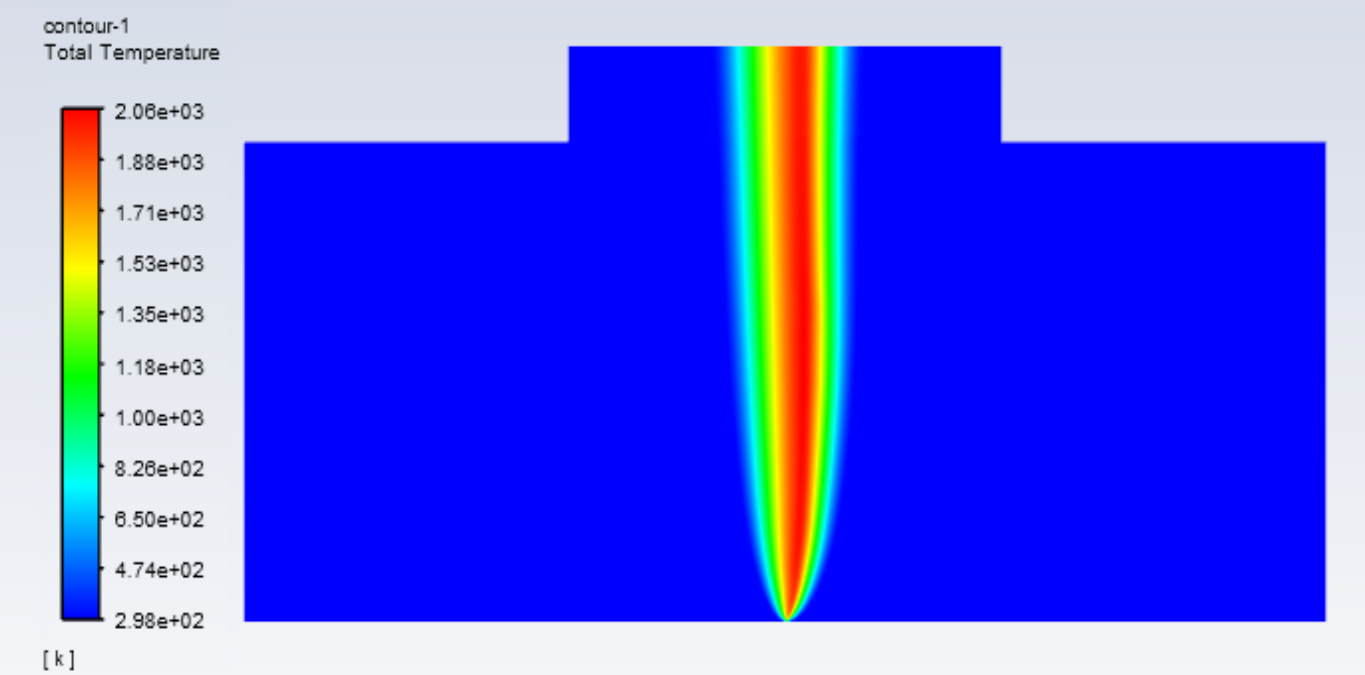

Figure 2: Example of a temperature contour plot

The figures above were created with a simplified 5-Step combustion mechanism (as discussed in the sections below). The diffusion mixing phenomenon that establishes a flame occurs at the interface of the fuel and oxidizer. The

shielding nitrogen encroaching on the side of the reacting area stymies the buoyant instabilities in the setup. At the top of the reacting area in the middle is a small slot designed for permitting the exhaust from the flame. Upon stabilization, this configuration yields a mixing layer a few millimeters thick, at least three times the thickness of the mixing layer typical of traditionally studied diffusion flame configuration [18]. The mixing layer thickness grows as the distance from the nozzles increase at a rate proportional to the square root of the product of time and velocity. Unsurprisingly, the time increases as the mixing layer interface grows farther away from the injection of the fuel and oxidizer. The only limitation that needs to be addressed in this configuration is mitigating any instabilities in the maintenance of the fuel-oxidizer mixture.

The main advantage in the PML configuration is the establishment of a large mixing layer thickness, relative to the other flame configurations. This will make the experimental probing for soot inception an easier endeavor. The simulations conducted in this study will help confirm this hypothesis. Implementing this configuration in the future and using it as a tool in studying soot formation will give the combustion community a better understanding of soot. This study aims at characterizing the structure of this flame with computational approaches and to pave the way to a broad use of this flame configuration for soot formation studies.

As alluded to before, soot has negative health and environmental impacts. Understanding its formation will continue to be vital when designing processes to ensure its existence is subdued.

3. Computational and Results Analysis Methods

3.1 Governing Equations

The modelling of the flow field, is performed by solving the equations of mass and species continuity and momentum and energy conservation with the Fluent software included in the ANSYS workbench. The solution of the continuity and energy conservation are necessary to predict any type of flow. The energy conservation must be accounted for because the flow of interest involves the transfer of heat, whereas the species conservation equation must be satisfied by the chemical reactions of multiple different species that take place in the considered flow. The combination of these fluid flow model equations are solved by Fluent to compute all of the relevant quantities in the simulation. For the USC2 mechanism, there are hundreds of such variables. These include, but are not limited to, the temperature, the velocity components, the concentrations of species, the pressure, the density, and the viscosity. The 5-Step mechanism [courtesy of Tianfeng Liu] considers 10 species as opposed to the 110 for the USC Chemical model [19].

3.1.1 Continuity Equation/Mass Conservation

One of the fundamental relationships satisfied by the flow field of a reactive flow is the continuity equation, which represents the conservation of mass. The principle of this equation is to ensure that the mass of a fluid remains constant throughout its flow [20]. The equation is given as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{\boldsymbol{u}}) = 0 \qquad (1)$$

Where ρ is the density of the fluid, *t* is the time, ∇ is the Del operator used with a product to calculate the divergence, and ***u*** is the velocity vector field of the flow. In the simulations for this study, the system was assumed to be at steady state. Thus, quantities do not vary with time. The first term of the equation can then be ignored. The density is modelled as constant for an incompressible fluid, so the second term of the equation can be rearranged and the equation is rewritten without the time-varying density term:

$$\rho(\nabla \cdot \vec{\boldsymbol{u}}) = 0 \qquad (2)$$

When the density is some constant, divergence of the velocity field is equal to zero. Additionally, the divergence of the velocity field integrated over the volume is the same as the flux of the field integrated over the surface of the volume. The equality of the volumetric integral of the divergence of the velocity field with the flux of the field integrated over the surface enclosing the volume is commonly referred to as

divergence theorem. In other words, for a given volume the combination of the sources gives the flux out of the region [21].

3.1.2 Momentum Conservation equations

The momentum conservation equation is another fundamental mode equation of fluid mechanics which can be solved by Fluent to compute the quantities of interest:

$$\rho\frac{\partial\vec{v}}{\partial t}+\nabla\cdot(\rho\vec{v}\vec{v})=-(\nabla\cdot\vec{p})+\nabla\cdot\bar{\bar{\tau}}+\rho\vec{g}+\vec{F} \qquad (3)$$

where $\vec{v}$ is the velocity vector, t is the time, $\vec{p}$ is the static pressure, $\rho\vec{g}$ is the weight of the fluid, $\vec{F}$ is the external force vector, and $\bar{\bar{\tau}}$ is the stress tensor. The stress tensor equals:

$$\bar{\bar{\tau}}=\mu\left[(\nabla\cdot\vec{v}+\nabla\vec{v}^{T})-\frac{2}{3}\nabla\cdot\vec{v}I\right] \qquad (4)$$

Where μ is the molecular viscosity and I is the unit/identity tensor. The general idea behind the conservation of momentum is that the momentum entering the system equals the momentum leaving the system plus the momentum accumulating into the system, the latter term being zero in the steady state condition analyzed in this study.

3.1.3 Energy Conservation

Another fundamental equation solved with Fluent in this study is the energy conservation equation. The approximate expression of this equation used in this study follows below:

$$\frac{\partial}{\partial t}\left(\rho C_p T\right) + \frac{\partial}{\partial x}\left(\rho V_x C_p T\right) + \frac{\partial}{\partial y}\left(\rho V_y C_p T\right) = \frac{\partial}{\partial x}\left(K\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(K\frac{\partial T}{\partial y}\right) + Q_v \quad (5)$$

Where C_p is the specific heat at constant pressure respectively, *T* is the temperature, V_i is the component of the velocity vector in a given *i* direction, *k* is the thermal conductivity and Q_v is the volumetric heat release due to the occurrence of chemical reactions. Note that the objective of this is to predict the flow field at steady state, thus the time derivative of the first term in the equation is equal to zero.

3.1.4 Conservation of Species

The final set of equations that allow modelling a reactive flow such as the one investigated in this study are the species transport equations which determine the concentrations of the multiple species composing the fluid. Each of the equations bases the balance of the i^{th} species characterized by a mass fraction, Y_i which can be readily converted to molar concentration.

$$\frac{\partial}{\partial t}(\rho Y_i) + \nabla \cdot (\rho \vec{\boldsymbol{v}} Y_i) = -\left(\nabla \cdot \vec{\boldsymbol{J}}_i\right) + R_{\boldsymbol{i}} + S_{\boldsymbol{i}} \quad (6)$$

Where R_i is the net rate of production of species *i* by the chemical reactions. For *N* number of fluid phase chemical species, this equation is calculated only *N*-1 times. Since the mass fraction of all the species must sum to one. Therefore, the *N*th mass fraction in the mechanism is yielded by the difference of one and the sum of *N*-1 previously solved mass fractions. To minimize the error in any solution, the last species should be the species with the largest mass fraction. Similar to the other equations, the time dependent quantities are not considered because the reactions take place at steady state.

3.2 Geometry and Mesh Generation

3.2.1 Geometry

The first step for solving the governing equations is to specify the region of space where the equations must be satisfied. In the planar mixing layer flame configuration investigated in this study, one wants to stabilize the flame at the interface of two vertical slot jets yielding from two adjacent inlets and containing oxidizer and fuel, respectively. The jets impinge into a horizontal stabilization plate downstream of the inlet reacting area while the flame products are exhausted through a slot machined in the downstream plate. The slot is used to avoid deflections of the mixing layer and to facilitate the exit of fuel exhaust. The reacting area, originally revealed in the introduction, looks as follows: The region in between

the inlets and outlets where one wants to compute the flow field and that is imputed to the Fluent solver depicted in Figure 3.

Figure 3: Computational area

The bulk of the reacting area has a width of 8 cm. The height of the portion of the reacting area without the slot is 5.5 cm. The length of the top is 4 cm while the top edges where the impinged nitrogen makes contact with the reacting area are 3 cm. The simulations in this study are completed in two dimensions. The surface area of the two dimensional reacting area is 43.2 cm^2. Finally, the width of the splitter is 0.4 mm.

3.2.2 Mesh Generation

After the geometry is imputed to the software, one must generate a computational mesh to discretize the governing equations. In this study the ANSYS

mesh preprocessor package was used to this end. The quadrilaterals meshing method was selected. This indicated that each element on the surface would be resolved into a quadrilateral. The global element growth rate, which is the ratio between the maximum dimensions of adjacent elements, was initially set to 1.05. Five distinct edges were selected to refine the mesh further. The first was the location of the splitter depicted in Fig. 4:

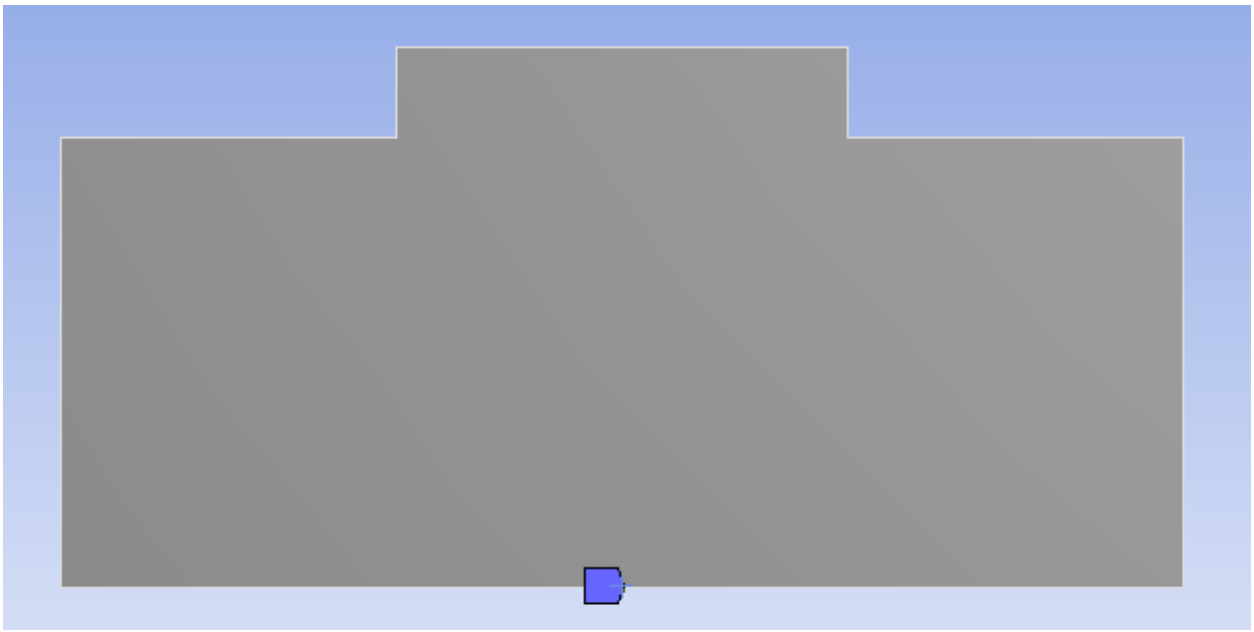

Figure 4: Location of the splitter

The maximum element size for this region was $1x10^{-4}$ m. The top edge depicted in Fig. 5 was refined.

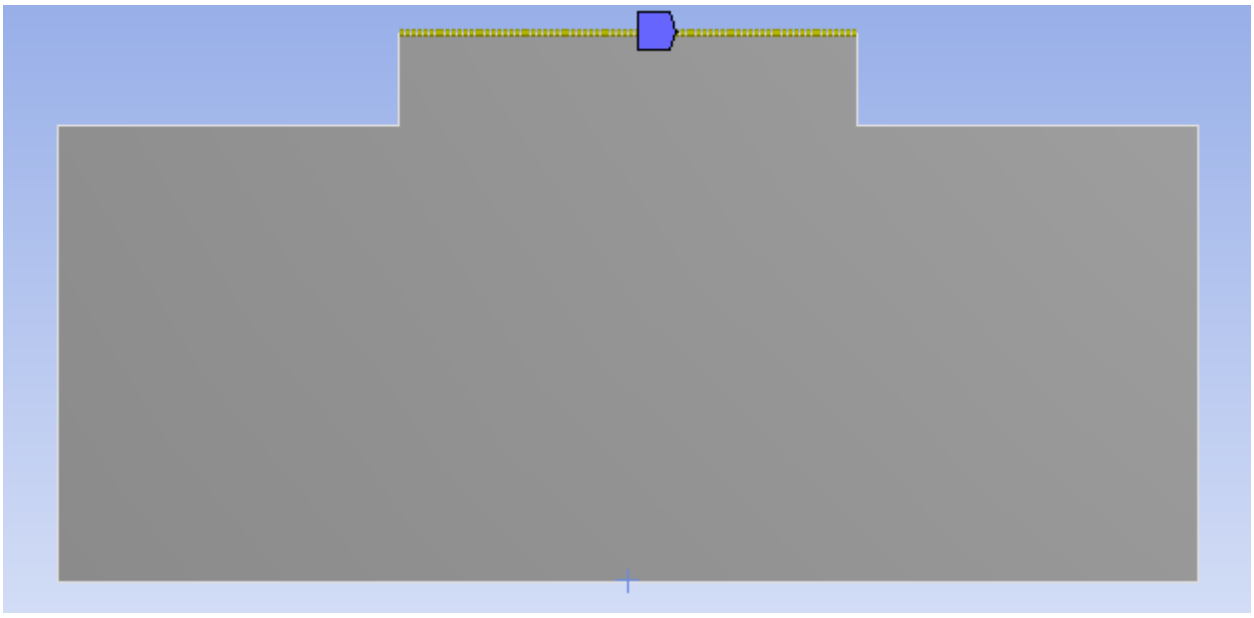

Figure 5: Outlet of the plate slot

The maximum element size for this edge was five times that of the splitter. The maximum element size was specified to be one millimeter at the inlets of the fuel and oxidizer, highlighted in Figure 3-4.

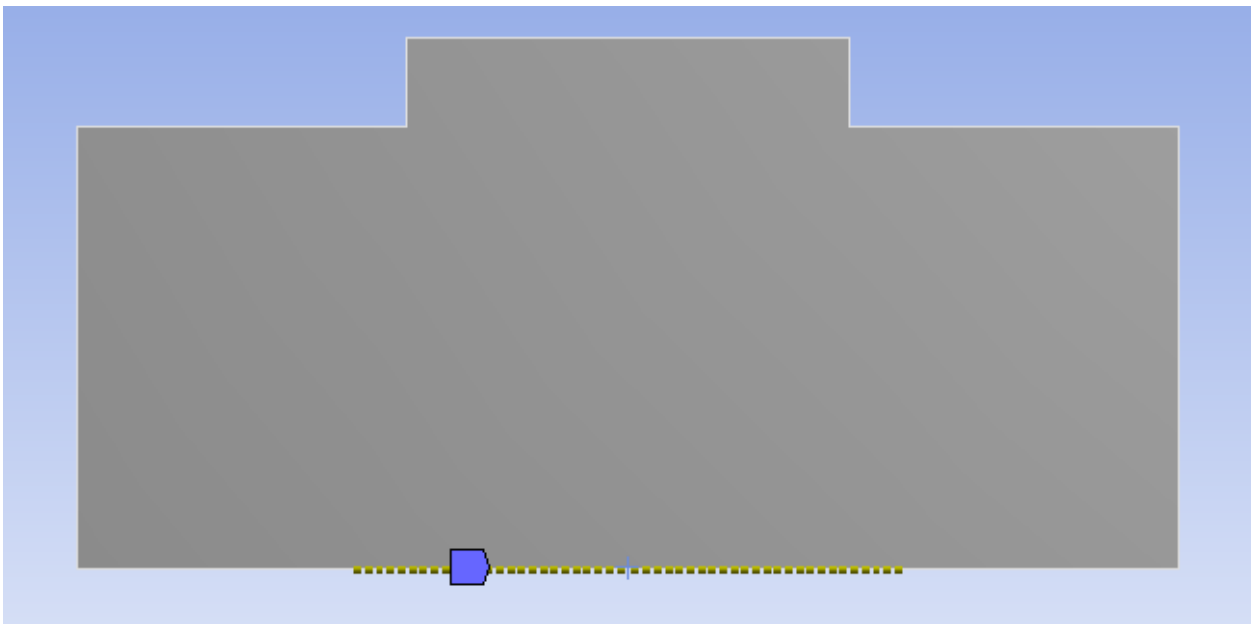

Figure 6: Oxidizer and fuel jet inlets

The nitrogen inlets and the side edges were also slightly refined. Even if these two areas of the reacting area are the least essential for analysis. This study is concerned with species and properties at the interface of the fuel and oxidizer and the size of the mixing layer. These phenomena are observed approximately in the vertical middle zone of the computational area. Thus the refinements for the nitrogen inlet and the side edges were the least robust. The maximum element size for the original mesh for these specified was two millimeters.

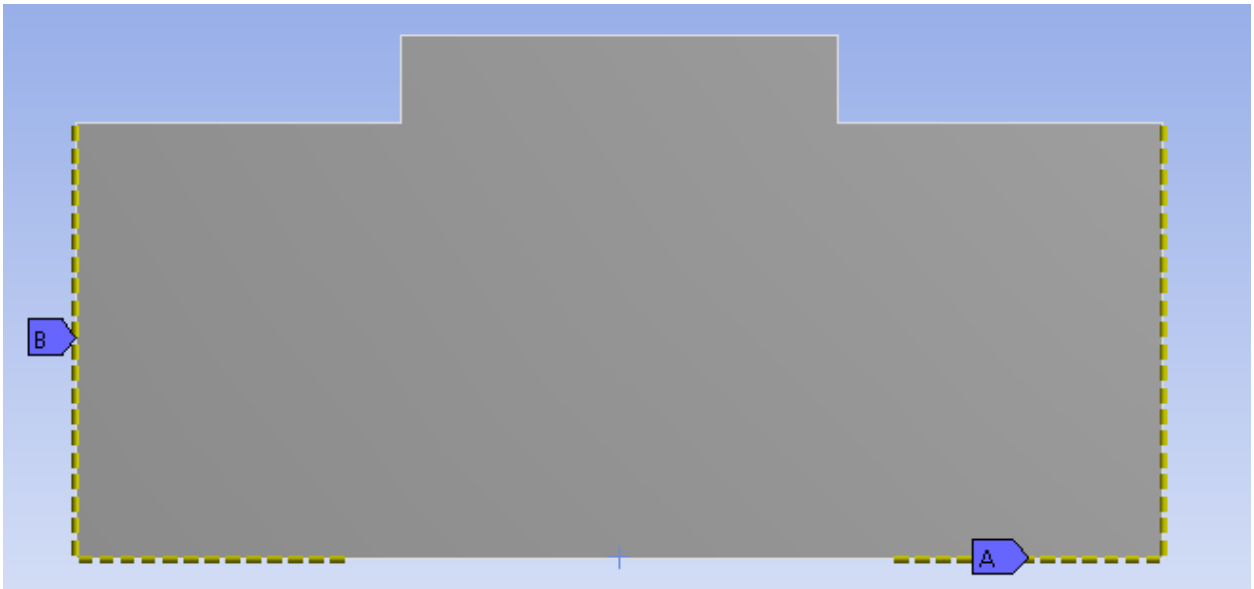

Figure 7: Nitrogen inlets and side outlets

3.3 Boundary Conditions

The governing equations can be solved only if they are accompanied by appropriate boundary conditions which the user must input for the computations to be initialized and converge to a solution. These boundary conditions set the

simulation in motion as the solutions for points in the middle of the grid depend upon the boundary conditions as the flow of fuel, oxidizer, and shielding nitrogen develops.

3.3.1 Pressure Inlet Boundary Conditions

The pressure inlet boundary condition enables the user to specify the pressure of the flow at the inlet of the system. This type of boundary condition is applicable to systems with compressible and incompressible flows. In this study, the pressure at the inlet for the fuel, oxidizer, and nitrogen was equal to the standard atmospheric pressure. Given the SI unit system was utilized in this study, the input pressure was 101,325 Pa.

3.3.2 Velocity Inlet Boundary Conditions

Velocity inlet boundary conditions grant the user the ability to provide the velocity of the flow at the inlet of the environment. The velocity inlet boundary condition can be used for incompressible flows but not compressible flows. Given this system is assumed to be comprised of incompressible fluids, the user can specify the inlet velocity.

The inlet velocity specification for an incompressible flow should not be at a location near an obstruction. If this condition is not ensured, the inflow stagnation properties will become highly non-uniform, leading to low fidelity solutions.

The inlet velocity boundary condition for each inlet needed to be specified in this study. This included the nitrogen inlet, the fuel inlet, and the oxidizer inlet. For each case the magnitude of the total velocity was specified. The direction was normal to the selected inlet boundary. The velocity of the flow at each inlet is 0.21 m/s.

3.3.3 Wall Boundary Conditions

For this study, plate walls (to the left and right of the outlet slot) and the splitter represent the physical boundaries of the system. Wall boundary conditions are a necessity in controlling what happens at the outer regions of the system. Without these conditions, the solution will not converge.

For both boundaries, the walls are considered stationary. So while the fluids are set in motion based on the combustion mechanism and inlet boundary conditions introduced above, the plate walls and splitter remain stationary.

Since this is a viscous flow, the no-slip boundary condition is applied to both the edges of the reacting area and the splitter. The no-slip boundary condition means that when interacting with any solid boundary containing the flow, the fluid will have zero velocity relative to the boundary. In this environment the walls are stationary. Therefore at the splitter and the edge of reacting area, the velocity of the fluid must be zero.

3.3.4 Temperature Inlet Boundary Condition

The user must specify thermal boundary conditions along the entire boundary of the domain. For this study we selected to assign the temperature at all fluid inlets. The temperature at the inlets and the solid walls was assumed to be all the same and equal the room temperature 298.15 K. At the top and lateral outflow portion of the boundary the temperature gradient in the direction normal to the boundary was assumed to be zero (fully developed flow).

3.3.5 Outflow Boundary Conditions

Outflow boundary conditions are leveraged by Fluent to simulate flow exits when the outlet flow velocity and pressure are unknown at the start of the problem. The conditions at the outlet are not manually defined; the software takes the information from the interior of the configuration and extrapolates the data required to yield the outlet conditions.

The outflow boundary conditions follow two rules: there is no diffusion flux for all of the flow variables and there is a mass balance correction for the mechanism as a whole. The zero flux for the outflow indicates that the conditions of the outflow nodes are based entirely on the interior of the configuration. This procedure from Fluent changes the velocity and pressure at the outlets in such a way that is consistent with the assumptions associated with fully-developed flow.

Fully-developed flows are those which remain constant in the direction of the flow. Note that the gradients of velocity and pressure might be non-zero at the outlet boundary. The zero flux condition only applies in the direction normal to the outlet.

Two outflow boundary conditions were specified in this configuration, the top outlet and the side outlets. Once these areas of the configuration are designated as the outlets, the user can define the flow rate weighting at each outlet. The flow rate weighting indicates to the solver how much of the total outflow goes through the selected boundary. In both cases, the flow rate weighting is one, which is the default value. When the flow is separated evenly amongst the designated outlets, then the flow weight readings do not need to be changed from the default setting. By leaving this default unchanged, Fluent will automatically adjust the flow rate fractions to generate the same flow fractions through each outlet. In the case of this study, there are two outlet conditions and the flow rates through each should be equal. Thus, Fluent takes the fact that the user does not change the default outflow settings and there are two outlet conditions and evenly splits the total flow between those two designations.

3.3.6 Species Boundary Conditions

To solve the problem, one must specify boundary conditions for the species transport equations. At inlets for fluid flow, the user can input the composition of

the species in the flow. This helps initialize the reactions dictated by the combustion mechanism. For this study the mass fraction was utilized as the method for specifying the species and amount of those species at the beginning of the simulation.

For the nitrogen inlet, the mole fraction of nitrogen is equal to one whereas the mole fractions of all other species is equal to zero. The mole fraction of ethylene (C_2H_4) is 0.33 at the fuel inlet whereas the mole fraction of oxygen (O_2) in the fuel stream is equal to 0.229. At both inlets nitrogen complements the total mole fraction to one since the mole fractions of all other species are set to zero. With these initial species specifications, the combustion mechanism can proceed with its characteristic chemical reactions. At the top and lateral outflow portion of the boundary the gradient of the mass fraction in the direction normal to the boundary was assumed to be zero for all species (fully developed flow).

3.4 Chemical Kinetic Mechanisms and Solvers

The USC2 [19] mechanism is the mechanism with which the applicable analysis will be performed ultimately. The USC2 mechanism includes reactions of 110 species of fluids which can be solved with the Chemkin solver in the ANSYS Fluent package. The simulations for the USC2 mechanism take about 20 hours to solve on the finest investigated computational mesh. To accelerate the

convergence a chemical equilibrium solver was the preliminary used to calculate and approximate solution which is subsequently used to initialize the computations with the Chemkin solver. A simple 5-Step mechanism [Tianfen Liu personal communication, Carbone et al. Combustion and flame 2015] was used in preliminary analysis to both gain familiarity with the software and wire the scripts to process the solution data. This mechanism considers only 10 species and can predict with reasonable accuracy the heat release rate (i.e. temperature and velocity field) of the flame. The computations performed with this simple mechanism take anywhere from five minutes to 30 minutes to complete, depending on the selection of the mesh size.

3.5 Adaptive Mesh

A method for improving the result's quality of a simulation is the application of an adaptive mesh to the geometry. The geometry has already been meshed when constructing the simulation; after importing or creating the geometry, the user is required to mesh that geometry. Implementing an adaptive mesh is an integral part of the setup after the geometry has been meshed. The adaptive mesh mechanism allows the user to manually specify which parts of the geometry should be calculated with more rigor and which regions do not require as much computational power. The former is referred to as the refinement criterion while

the latter is the coarsening criterion. The user can manually define regions or boundaries that need to be treated differently in the cell registers. In this case, the refined region is that where the half width temperature and max temperature streams are located. Correspondingly, this is the location of the mixing layer. Here is the region where the refinement occurs, depicted in Fig. 8:

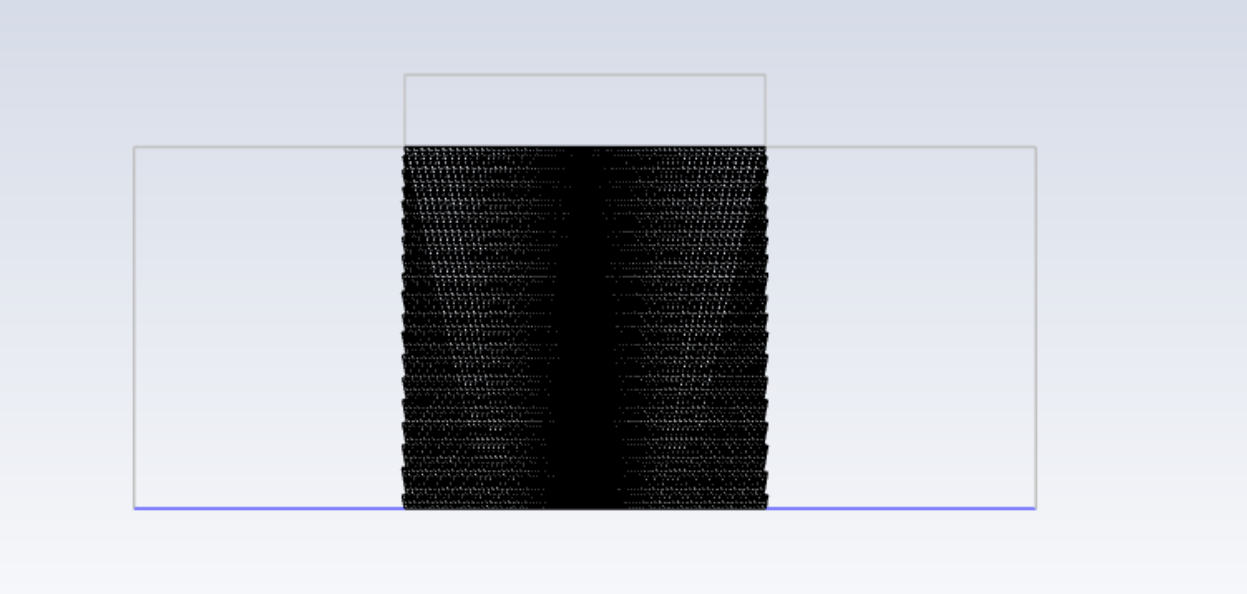

Figure 8: Mesh refinement region

The dimensions are as follows: the horizontal coordinate is between 0.02 m and -0.02 m and the vertical coordinate is between 0 m and .04 m. To define the coarse region, one can specific all the nodes that are not in the refinement region. In Fig. 9 are all the node in the coarsening region:

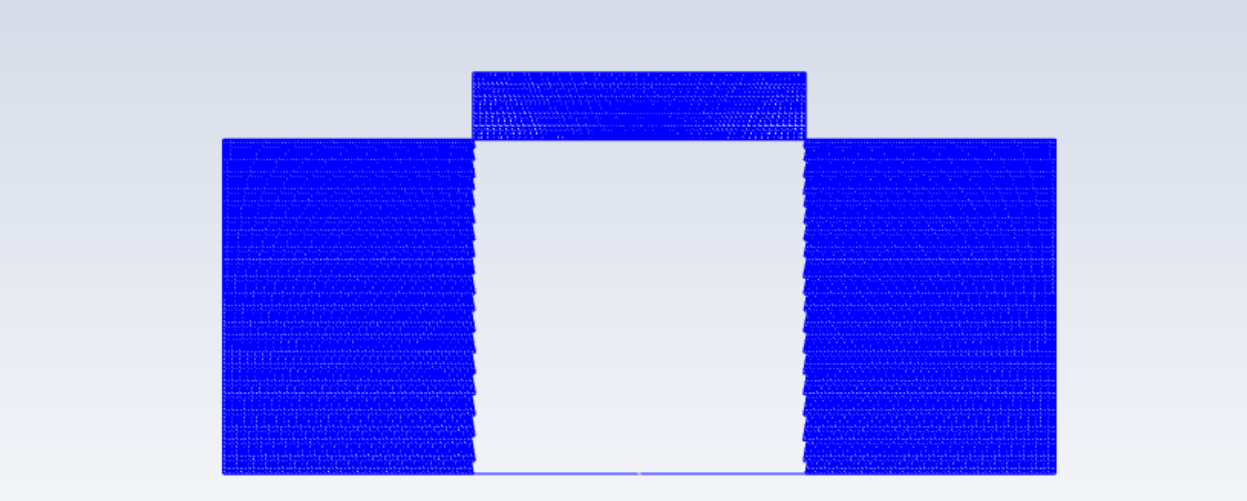

Figure 9: Mesh coarsening region

When running the simulations, the solver will calculate the properties of the nodes colored blue with less iterations. In turn, the solver will run the necessary calculations for these nodes with additional iterations, which the user can specify in the adaptive mesh dialogue box. The adaptive mesh is a useful tool when computing the solutions of models that consider complicated chemical mechanisms with a large number of species and reactions involved. Indeed, by choosing which nodes require more attention and which require less, the computational time is cut down. Since the mixing layer develops in the region denoted by the black nodes, the solver should focus on that region. In the future, when running calculations with the USC2 mechanism, users can cut down on computation time and increase the number of simulations to generate more data in a given timeframe.

Another way of adapting the mesh is to manually change the element sizes and growth rates of each part of the mesh. In Fig. 10 and Fig. 11, one can notice the differences in the two meshes.

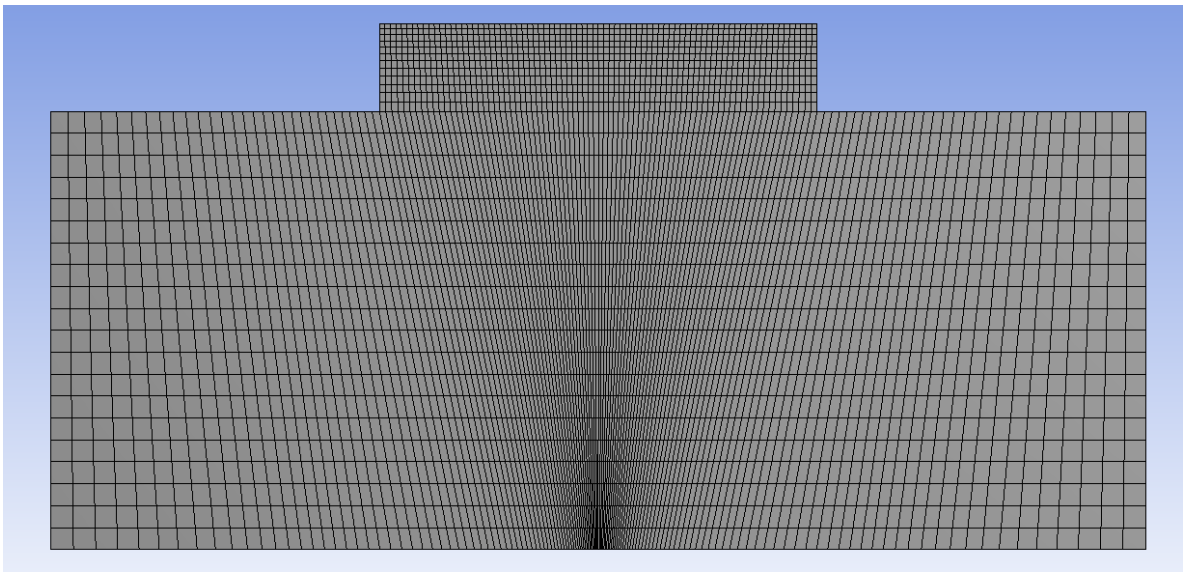

Figure 10: Original mesh

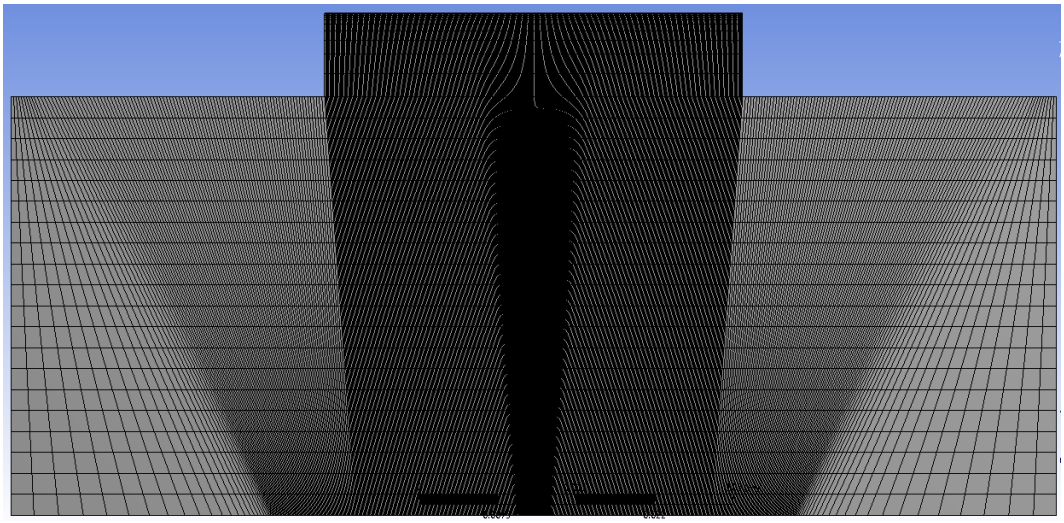

Figure 11: Adaptive mesh

3.6 Initializing Chemkin Solutions

ANSYS facilitates the initialization of simulation with previous simulation results. When running and analyzing the USC reaction mechanism with 110 species, having the ability to initialize the solution will assist in cutting down on computational time. Given the scope of the simulation, this is an essential function that mitigates the computational power, which would otherwise be more time when running the more robust simulation.

Before initializing the more complex mechanism the initialization technique must converge with the solution of an uninitialized simulation. The impetus behind utilizing this technique is not too yield a more accurate solution, but to obtain a solution with more alacrity while maintaining the integrity of the results. Thus, conducting a convergence test for the initialized solution by comparing its results to those from the uninitialized solution in the 5-Step mechanism with only 10 species is necessary. The pressure values are compared the between the two solutions at each vertical cut in the mesh. Pressure is the ideal quantity with which to test convergence due to its relative sensitivity (with respect to the other quantities measured in ANSYS Fluent simulations) between solutions.

To initialize the solution, first a simulation is run with the chemistry solver in Fluent set to “Relax to Chemical Equilibrium”. Given the assumptions leveraged in

this solver, this solution is less computational laborious compared to that of the Chemkin solver. The equilibrium solution needs to be checked for convergence (based on the value of the continuity equation residual reaching a standardized value of 10^{-4}). Upon the completion of the simplified solution, the results were saved as an IP file, a file type that Fluent can read as the basis for simulation initialization. I generated the file using the interpolating functionality in Fluent. After exporting the solution data, the chemistry was switched to the solver "Chemkin- CFD Solver" to create more physically reliable simulations. First the simulation was run by automatically initializing the solution and exporting the necessary data as a CSV file. Then, the interpolate function was used to export the equilibrium solution data and imported the equilibrium data into the Chemkin simulation, which would initialize the Chemkin solver with the results from less robust simulation method. Similarly the initialized Chemkin solution was exported as a CSV file.

Post processing and comparing the two data sets using the R programming language in RStudio, an integrated development environment for the aforementioned programming language, allowed for a test of the initialization function's merit. The code wrote to process the data is below the plot comparing

the two solutions. The following is a solution comparison of the pressure values at each value for the x coordinate, separated by each y value:

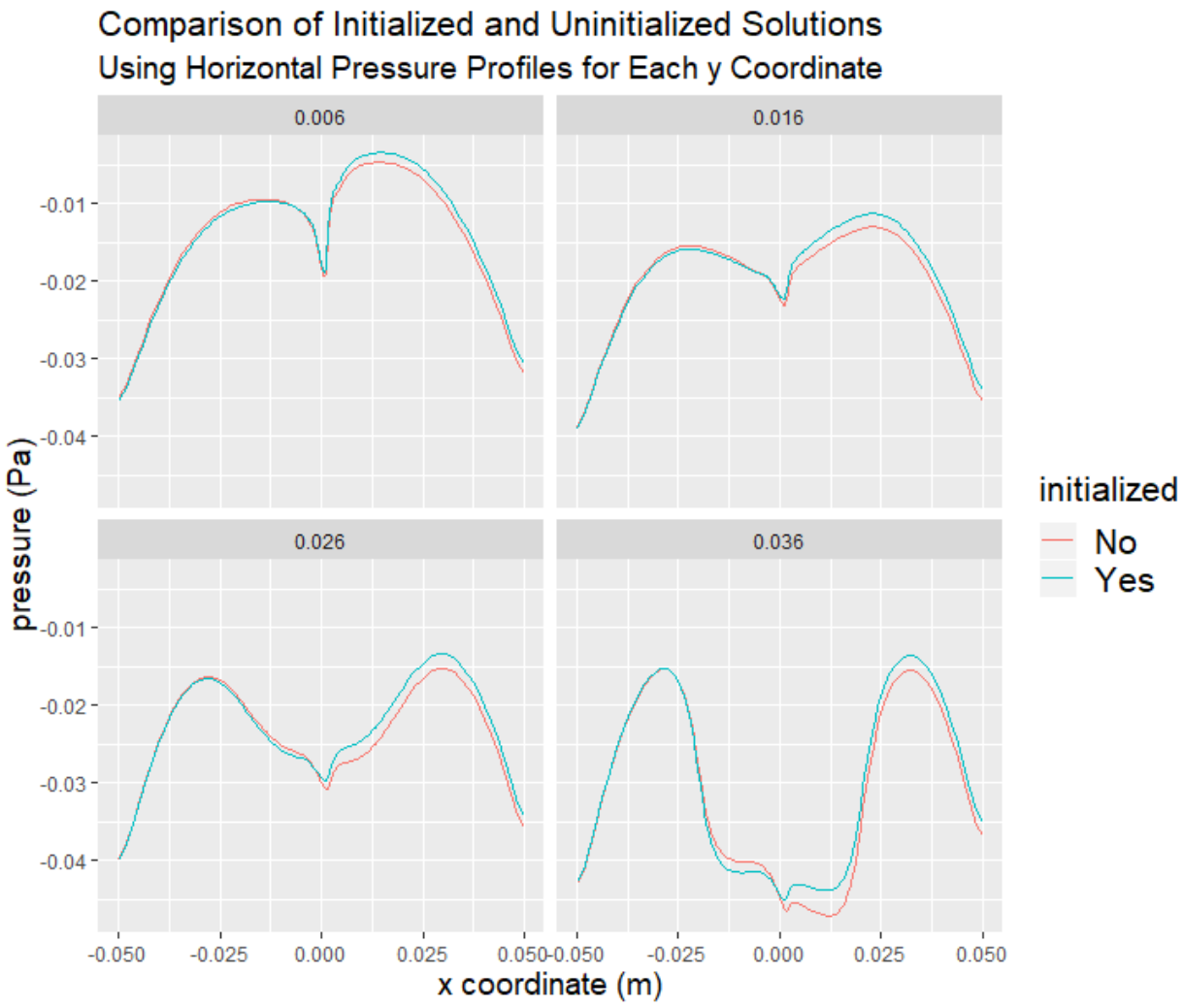

Figure 12: Initialized versus uninitialized solutions

As one can see, there is agreement in the two solution types. Furthermore, the relative error was checked between the two solution types for the pressure values at each horizontal coordinate:

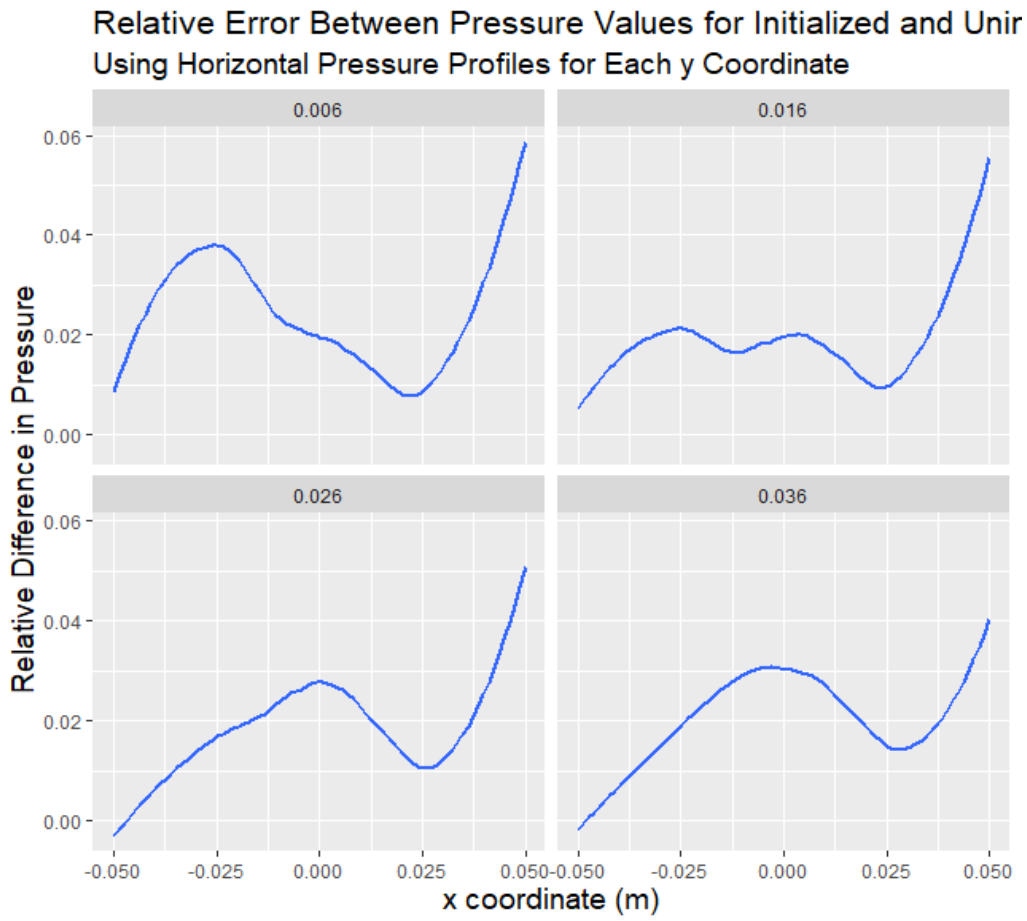

Figure 13: Relative error between initialized and uninitialized solutions

The relative error is small, no more than six percent at any point on the grid. By verifying that the solution will behave similarly with the initialization method explained before, one can utilize it to run the more complex simulations with 110 species. The temperature profiles were observed as the mixture developed to check if the mixing layer thickness increases as time goes on and that the maximum temperature for a given vertical coordinate also increases as the distance from the inlet increases:

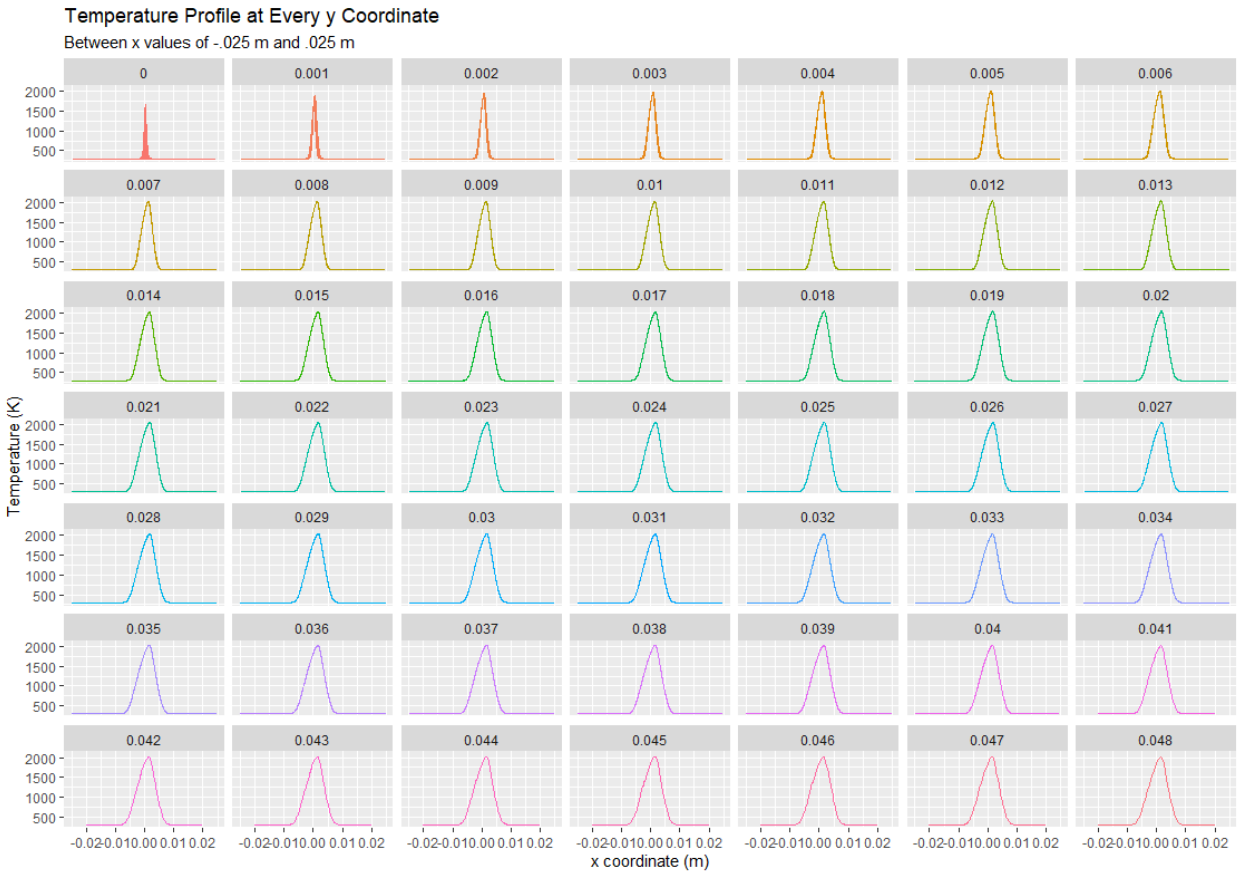

Figure 14: Temperature profiles at each HAB

The initialization method was determined to be viable. For the USC2 mechanism, this was required, as opposed to the 5-Step mechanism. The change in scale of the number of species necessitated the need to manually initialize the simulation with a file. The USC2 mechanism was run with the solution relaxed to chemical equilibrium for 1500 iterations. The data was saved and then subsequently used to initialize the simulations utilizing the Chemkin solver for all the USC2 simulations for the remainder of the study.

3.7 Streamline Functionality in ANSYS

Fluent has a function where users can identify streamlines. This is useful for understanding how particles develop as time increases in the solution. The streamline at the interface of the fuel and oxidizer was picked out. A XML file was written that could be processed in CFD-Post, where a CSV file can then be written with all the relevant information about the streamline, most notably its position, the time when it reached a given position, the velocity components at each point, the temperature at each point, and the composition of the mixture at each point. The temperature and time are especially relevant given those quantities can be used to normalize the temperature values and horizontal coordinate respectively for a horizontal profile at any vertical coordinate. First the location of the streamline was exported from Fluent to see check its location on the grid:

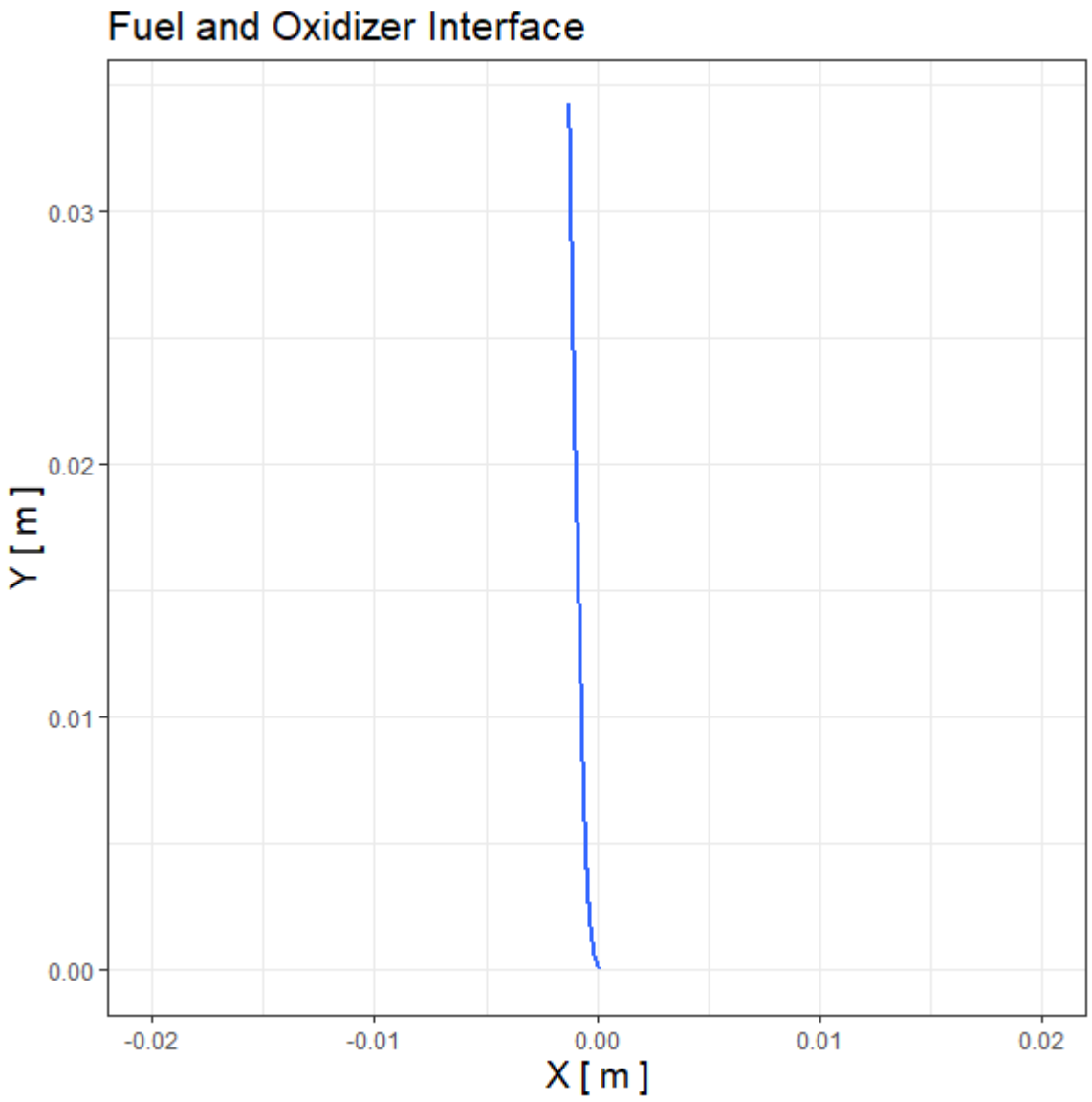

Figure 15: Interface streamline

Then the interface was visualized with the temperature to understand how the temperature changed as time elapsed:

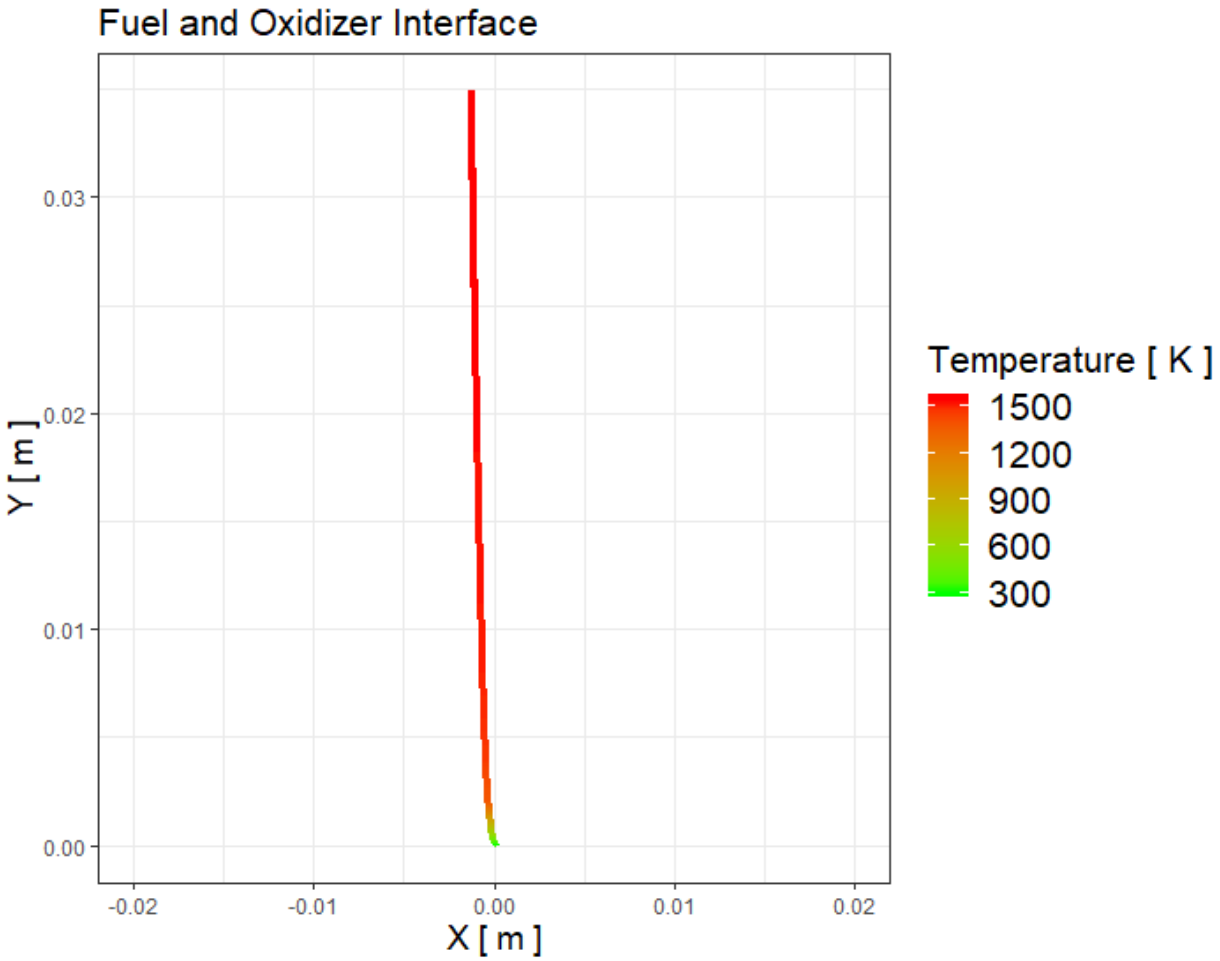

Figure 16: Interface streamline temperature

I processed the data myself, identifying the locations of the fuel and oxidizer interface and the time elapsed for each point. The streamline function in Fluent is rigid, in that narrowing down the solution to identify the streamlines of interest is tedious. The function becomes of great use, on the other hand, as a method of validating my own data processing code. If the results from the ANSYS function match the results from my code, one can be confident in how I process that data going forward.

4. Results and Discussion

4.1 Mesh Convergence

The 5-Step mechanism was used to provide the basis for the methods to properly analyze the USC2 mechanism. So the techniques described were then applied to the USC2 mechanism.

4.1.1 Mesh Convergence of 5-Step Mechanism

Before being confident in the solutions yielded in fluent, one must ensure that the numerical results are mesh independent. Such independence can be proved showing that the results do not change or become more refined based on the size of the mesh. To assess the convergence of the results towards a mesh independent solution, simulations were run using the simplified five-step chemical mechanism and four different meshes. These meshes were built with an increasing number of nodes. Operatively, the simulations varied with respect to the element size (i.e. the distance between nearby nodes) of each mesh, which correspondingly yielded a different number of nodes in each case. The coarsest mesh used in the initial computations has a maximum element size of $4 \cdot 10^{-3}$ m, which resulted in 3933 nodes. Successive mesh refinements resulted in maximum element sizes of $1 \cdot 10^{-3}$ m with 7532 nodes, $5 \cdot 10^{-4}$ m with 20,493 nodes, and of $4 \cdot 10^{-4}$ m with 30,381

nodes, respectively. Finally, the finest mesh tested has elements that were size at $3 \cdot 10^{-4}$ m and had a total of 51,134 nodes.

Upon creating all five meshes, the simulations ran once for each mesh and the solution data was exported from fluent. Using scripts developed with R in RStudio, the data was processed as necessary and began to analyze the results. For an initial check, before more in-depth analysis, the maximum temperatures along various horizontal profiles were observed at several vertical coordinates (Heights Above the Burner, HAB) and the results are shown in Fig. 17. This plot gives preliminary indication about the convergences of the results toward a mesh independent solution. I conducted this first check the general behavior of the solutions before putting time into more robust analysis. As expected the values of the maximum temperatures approach asymptotic values as the number of nodes constituting the mesh increases.

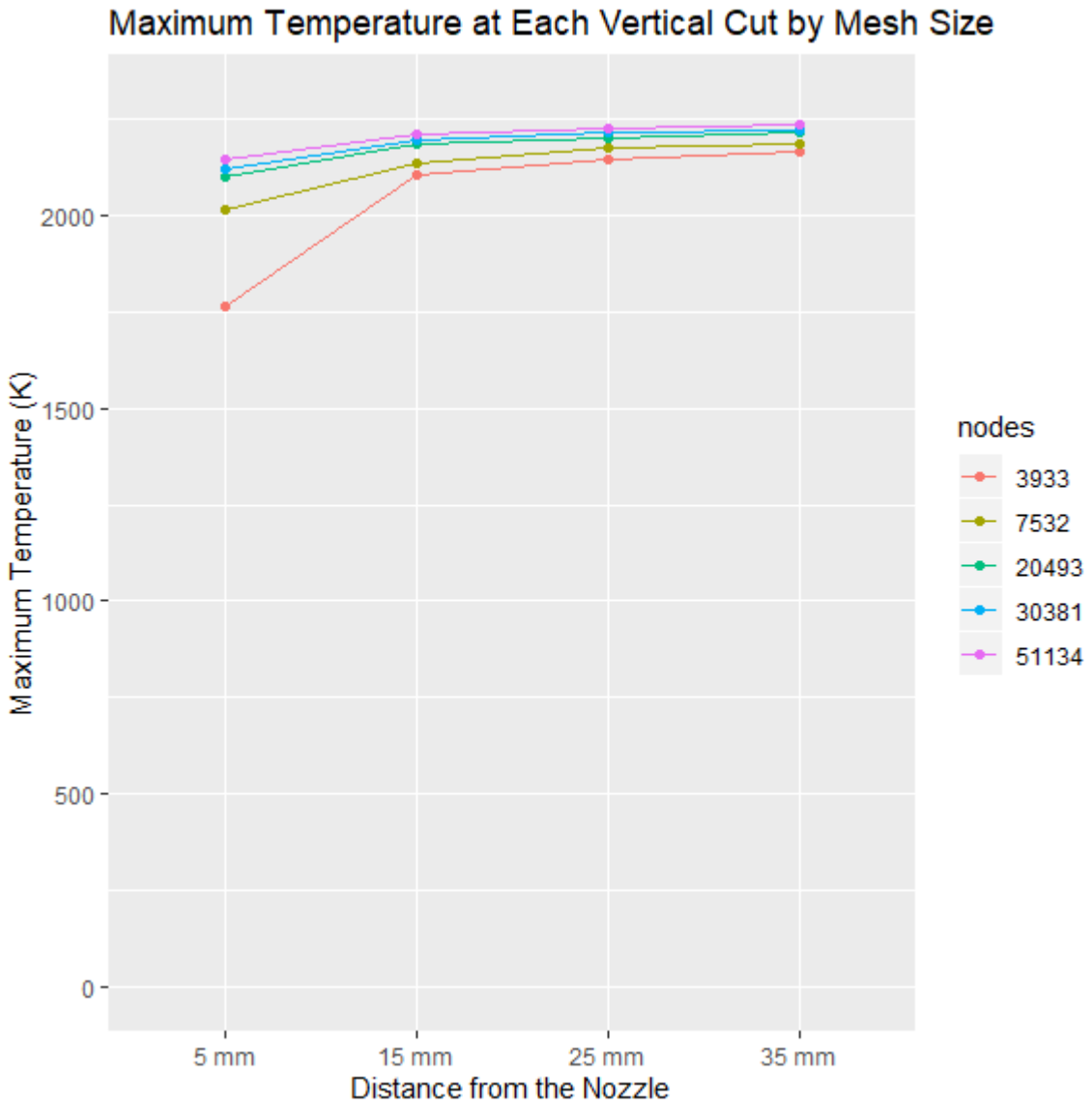

Figure 17: Maximum temperature at select HABs

Next more robust analysis was conducted on a few different variables from the solution. The visualization in Fig. 18 represents the distribution of static pressures on the grid at selected vertical coordinates.

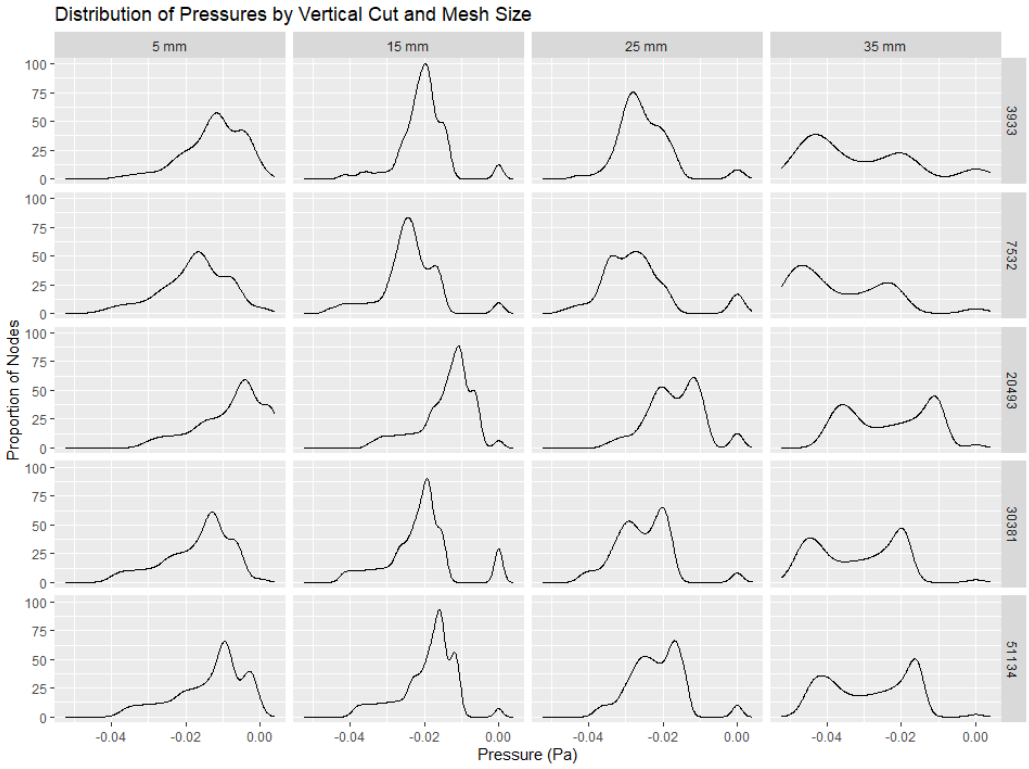

Figure 18: Pressure distributions at select HABs

When looking at the maximum percent difference in pressures between the two finest meshes, the result comes out to be 7.81%. As one can see, the distribution of pressure values converge to a single shape as the number of nodes increases in the mesh. Similarly, the same type of analysis was performed on selected horizontal profiles for both horizontal velocity in Fig. 19, temperature in Fig. 20, and the mole fraction of carbon monoxide in Fig. 21.

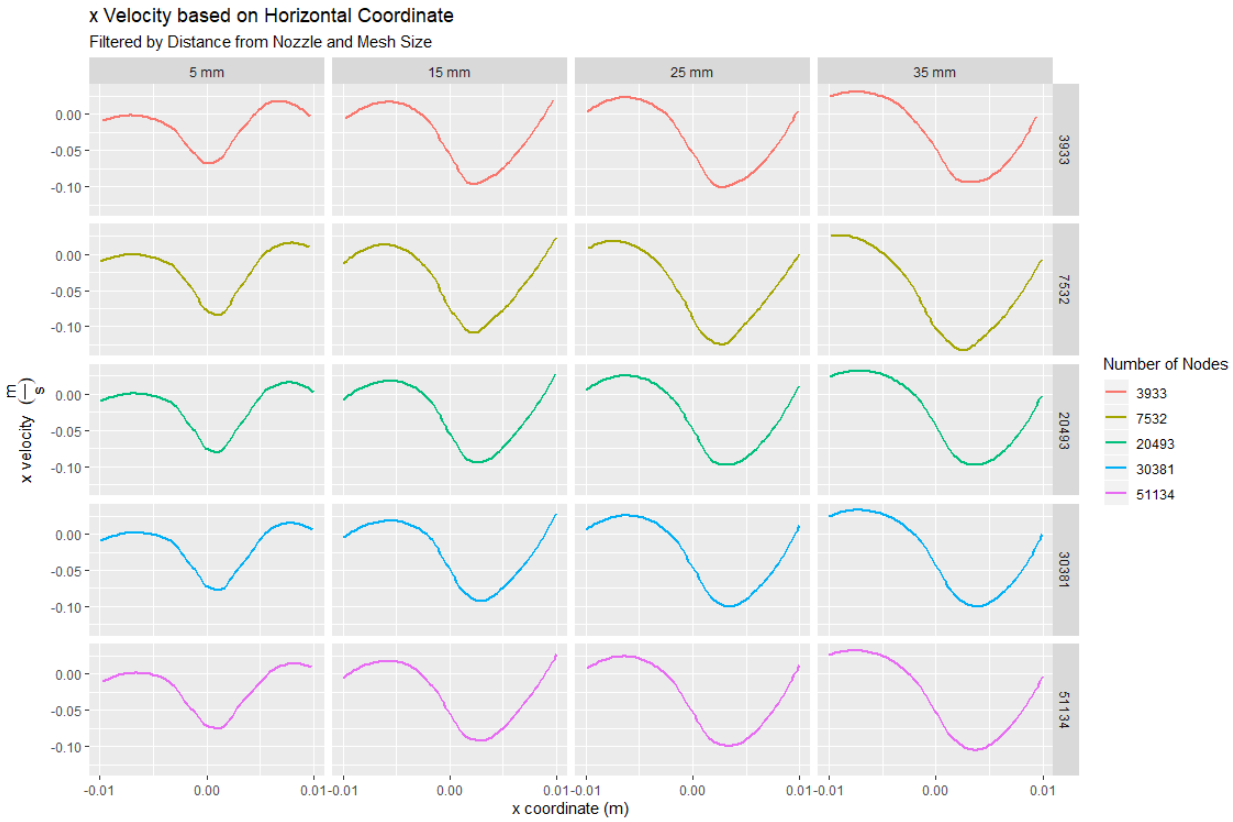

Figure 19: Horizontal velocity profiles

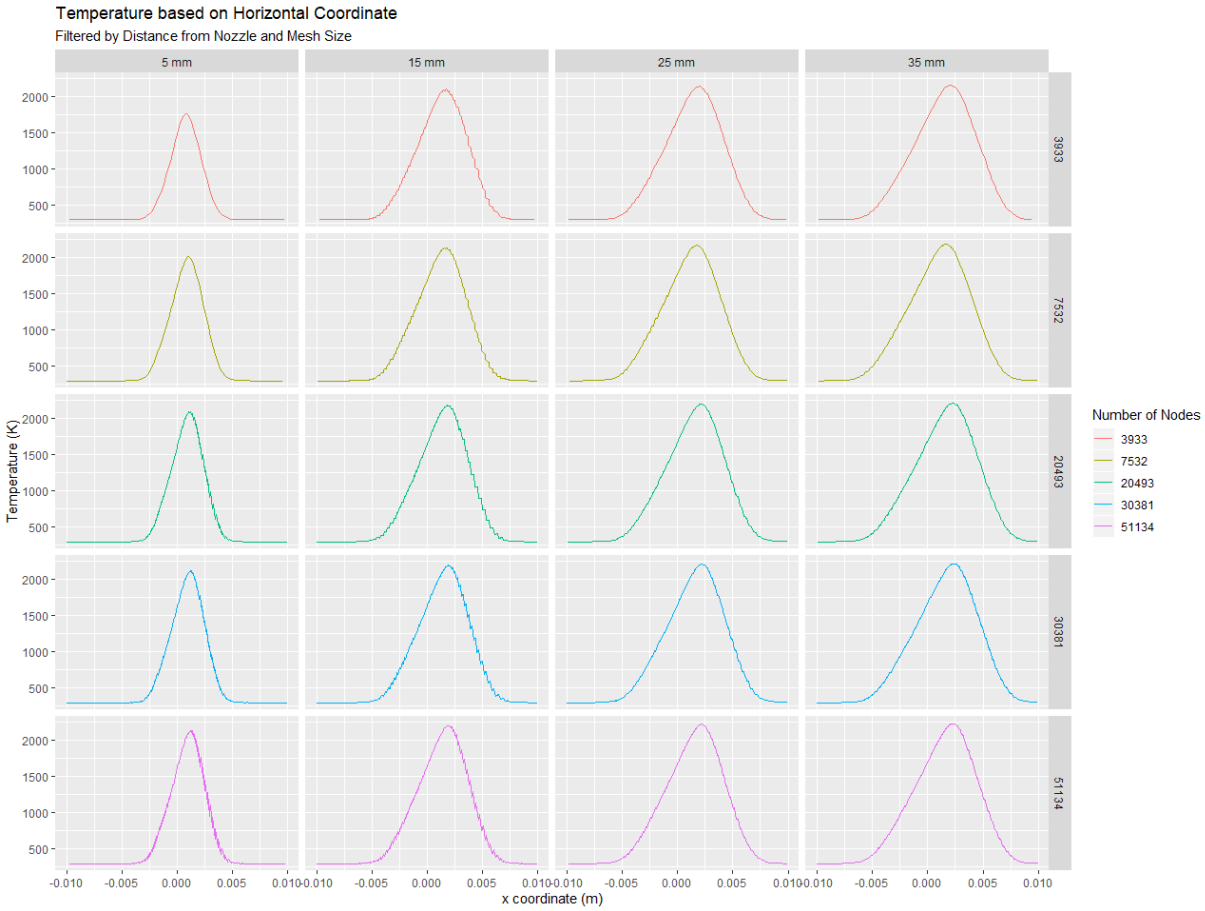

Figure 20: Temperature profiles at select HABs

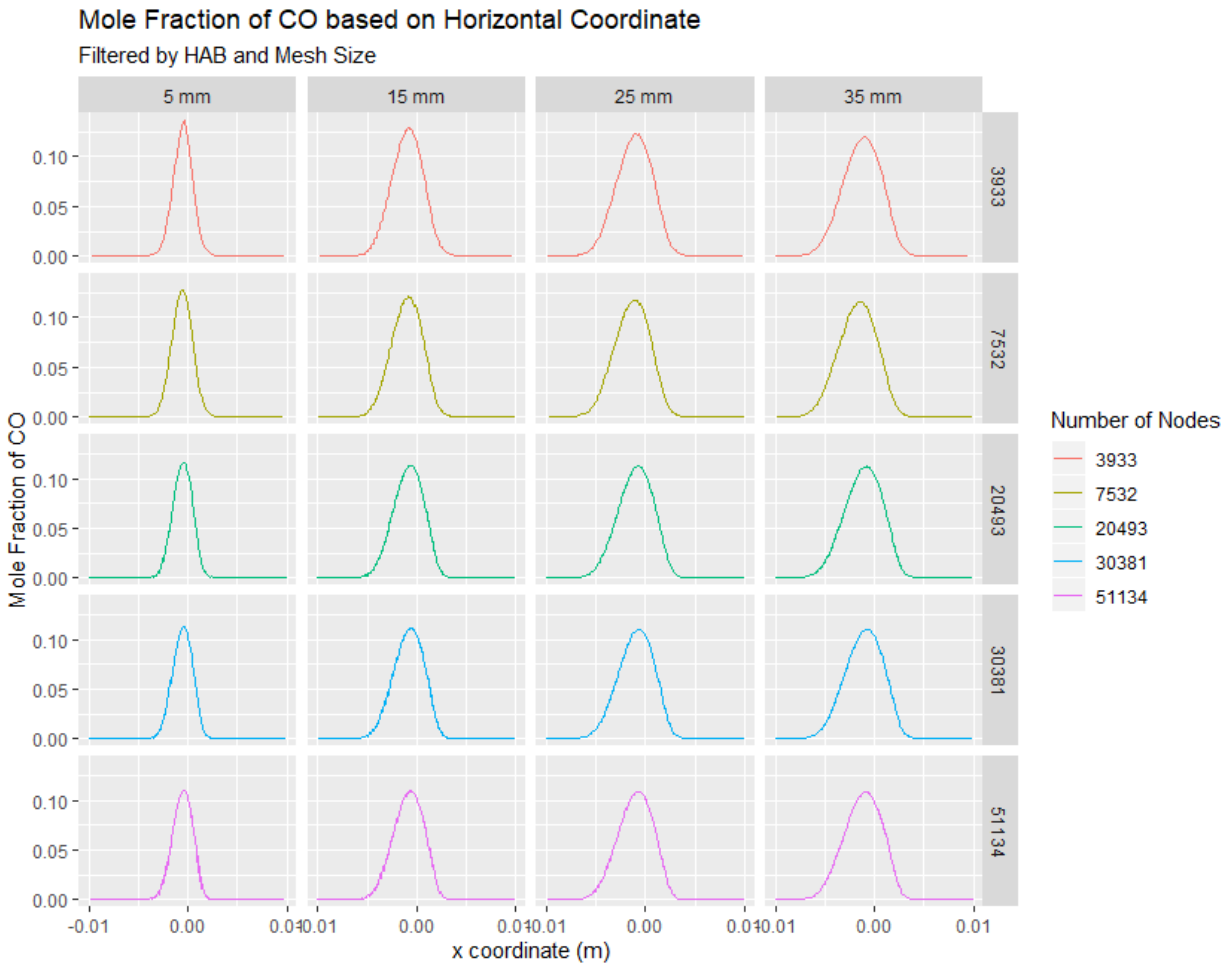

Figure 21: CO concentration profiles at select HABs

Based on these profiles of a select few properties, one should be confident in the convergence of the results toward the mesh independent solution for the model generated using the simplified 5-step chemistry.

4.1.2 Mesh Convergence of USC2 Mechanism

Similar to the mesh convergence analysis with the 5-Step mechanism, the results obtained with the USC2 mechanism were compared upon implementing mesh successive refinements. The same R scripts were used, except the data files for the

USC2 mechanism were substituted for the 5-Step mechanism files. Figure 22 shows the maximum temperatures in the horizontal profiles at each vertical coordinate for meshes with a different number of nodes (see legend).

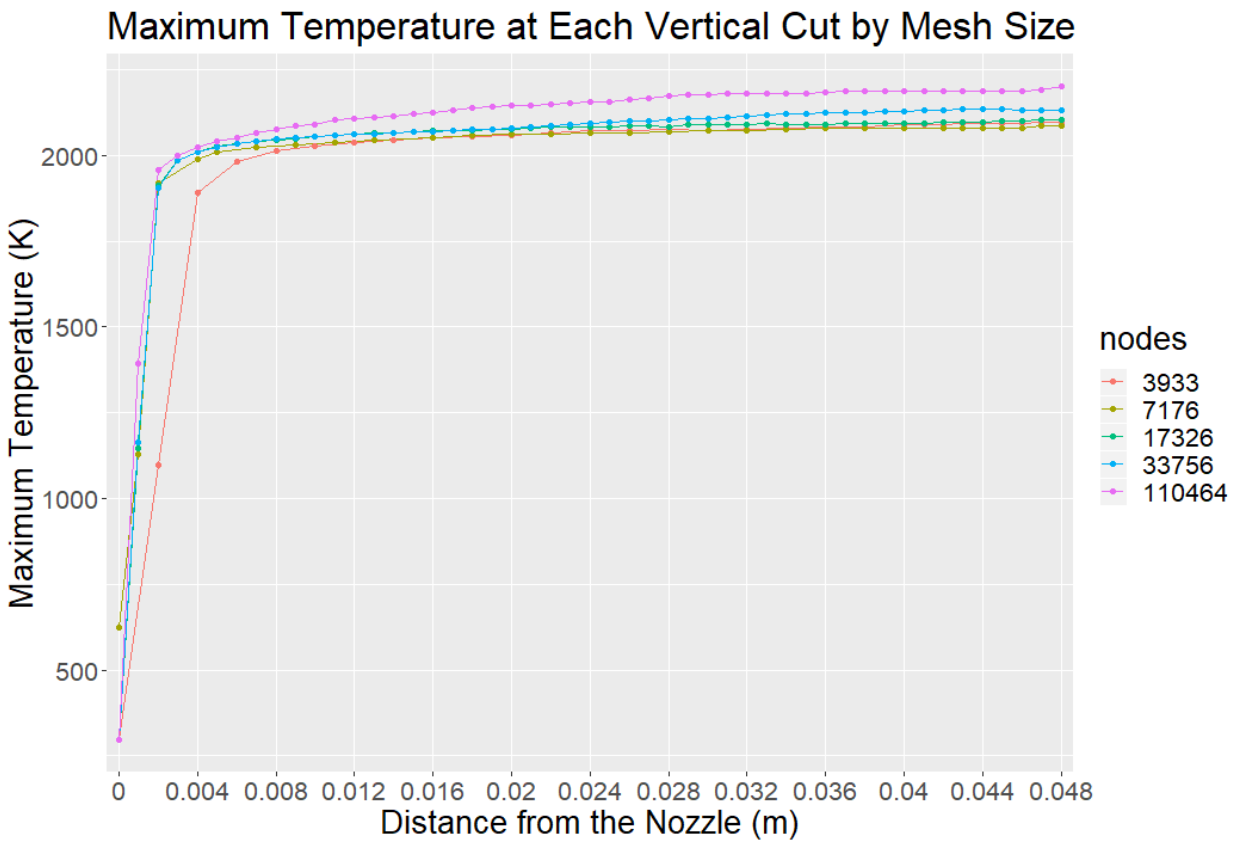

Figure 22: Maximum temperature at each HAB

Figure 23 and 24 show the horizontal profiles of pressures and horizontal component of the velocity computed at select HABS, respectively.

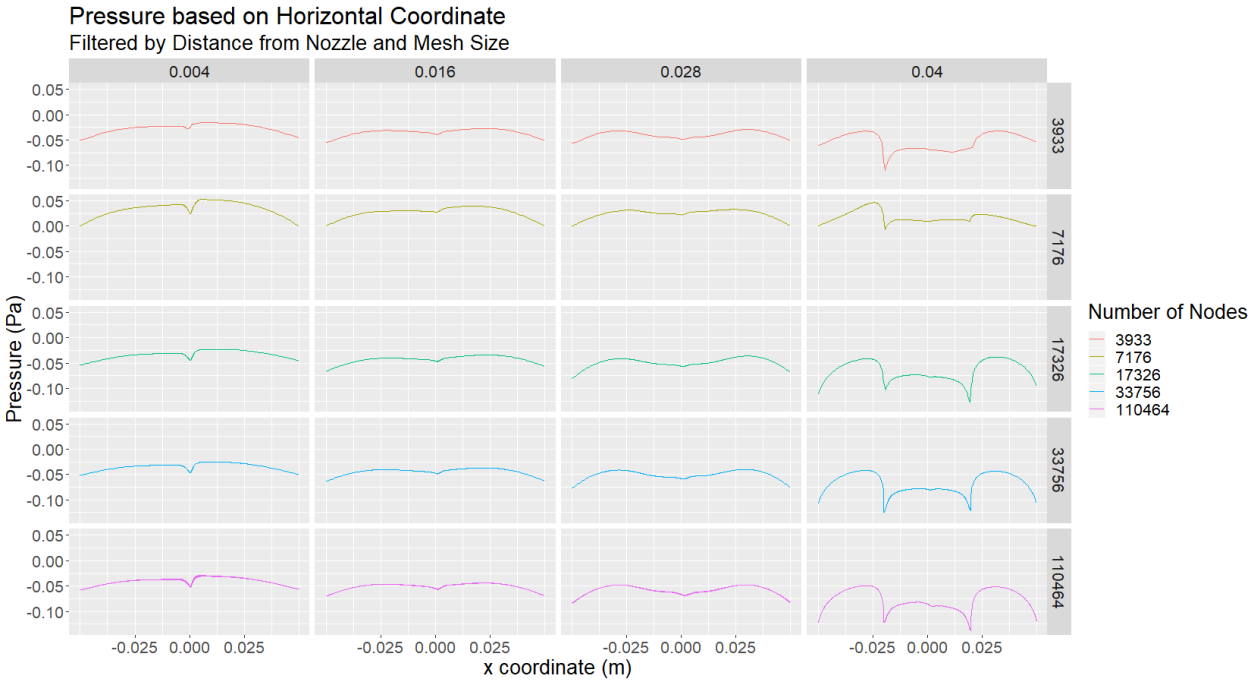

Figure 23: Pressure profiles at select HABs

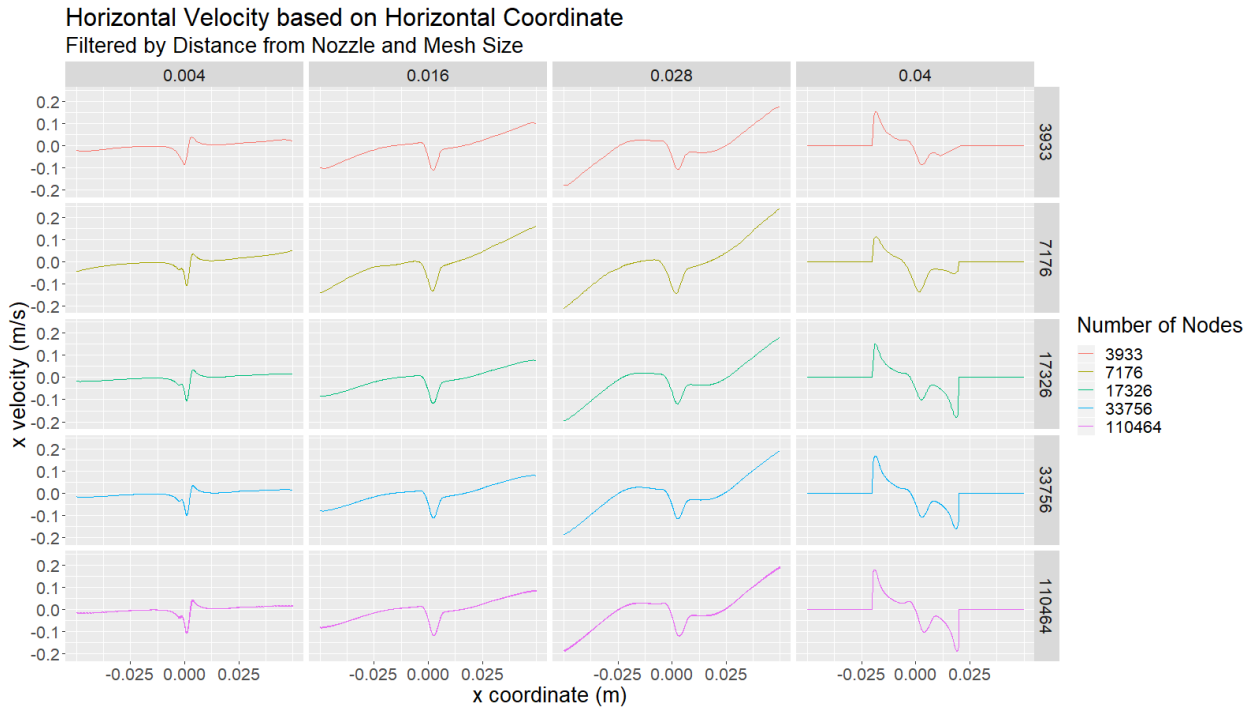

Figure 24: Horizontal velocity profiles at select HABs

Finally, Figures 25 and 26 depict the horizontal profiles computed at select HABs of temperature and concentration of benzene (C6H6).

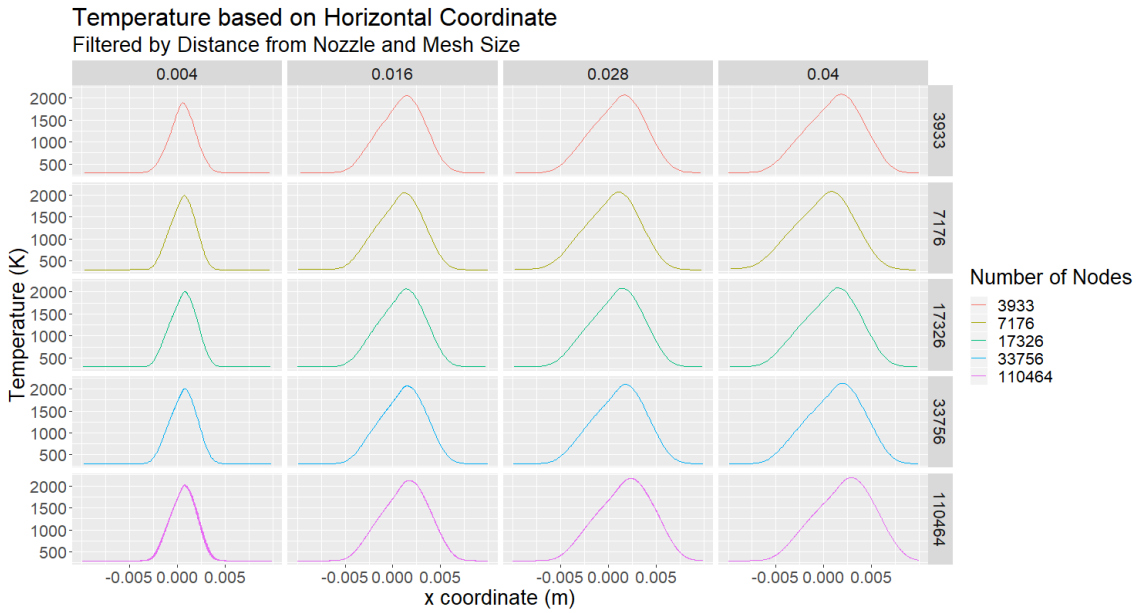

Figure 25: Temperature profiles at select HABs

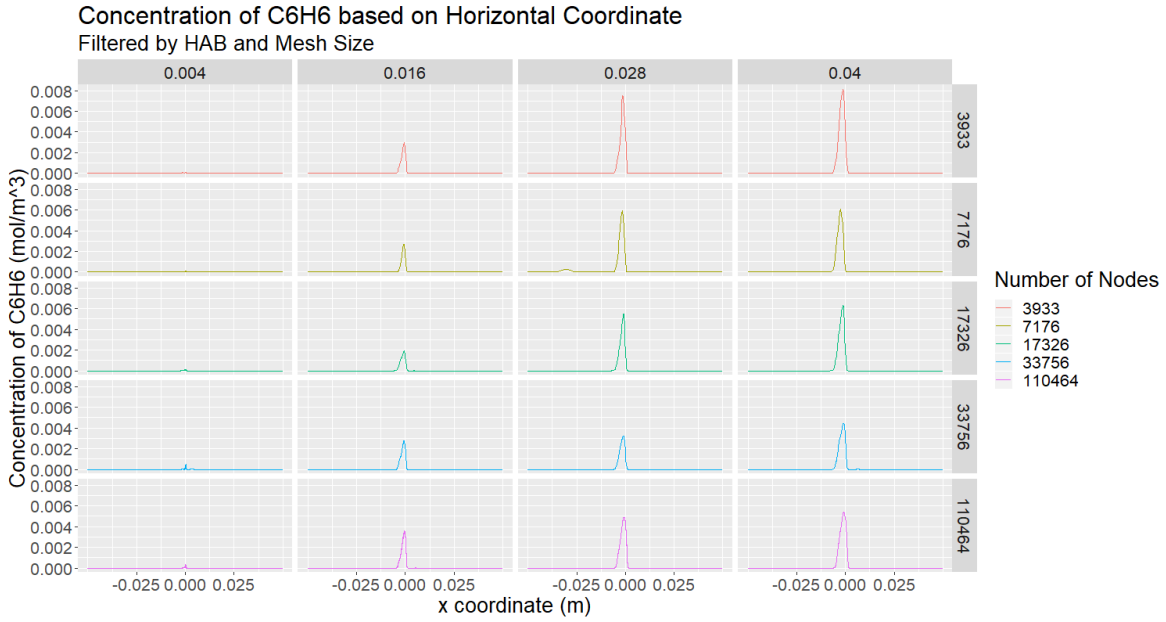

Figure 26: C6H6 profiles at select HABs

It appears that even if mesh refinements have minimal effects on the computed value of the flow field (e.g. pressure, velocity, and temperature), they significantly impact the computed concentration of benzene so that additional mesh refinements are needed to obtain grid independent results with respect to this concentration of species. To further investigate this finding, a similar plot was made to that with the maximum temperature, but with C6H6 concentration in Fig. 27.

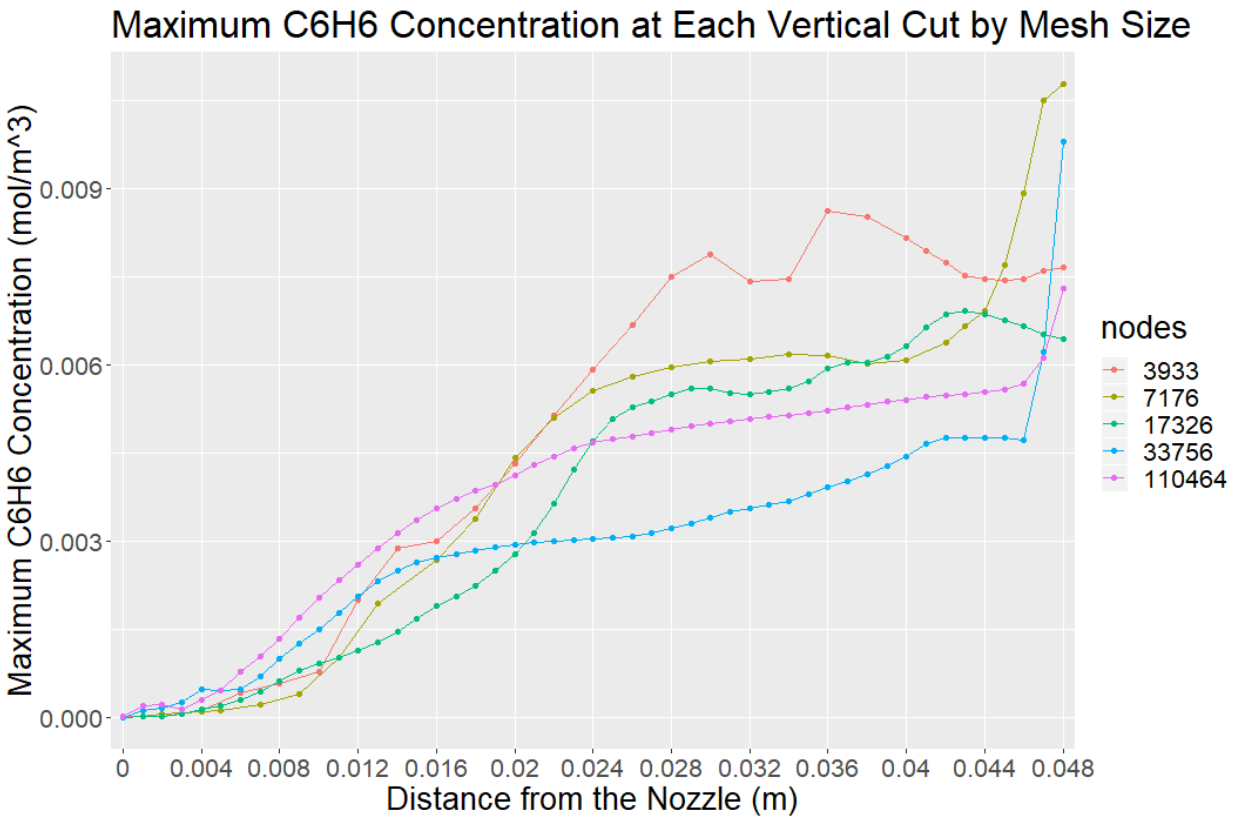

Figure 27: C6H6 concentration at each HAB

Even with the dramatically refined mesh, the C6H6 concentration does not seem to achieve mesh independence. Next, for the finest mesh, the resolution of the

solution was changed based on the convergence criterion for the C6H6 residual. The results are in Fig. 28.

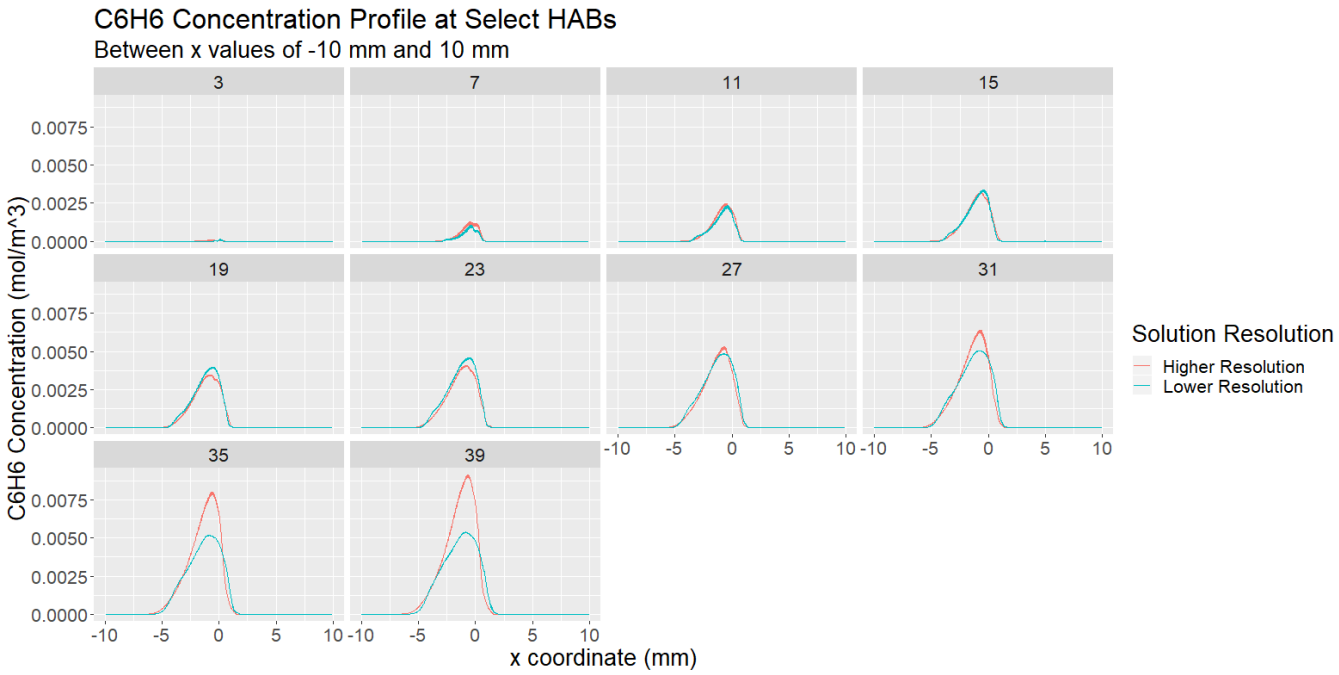

Figure 28: Comparison of C6H6 profiles with different solution resolutions

When the convergence criterion is made stricter, the concentration of benzene changes as the time within the simulation increases.

Given the scope and resources of this investigation, more fine meshes nor more strict convergence were tested due to computational costs and time constraints. However, the general trend regarding the increase in C6H6 concentration is similar for each mesh size. Specific results yielded from C6H6 concentration should not be considered as perfect encapsulations of this system when physically realized, but the general ideas and trends are still helpful. To

further check the validity of the results with respect to their ability to predict the physical setup of the PML, the maximum C2H2 were obtained in Fig. 29.

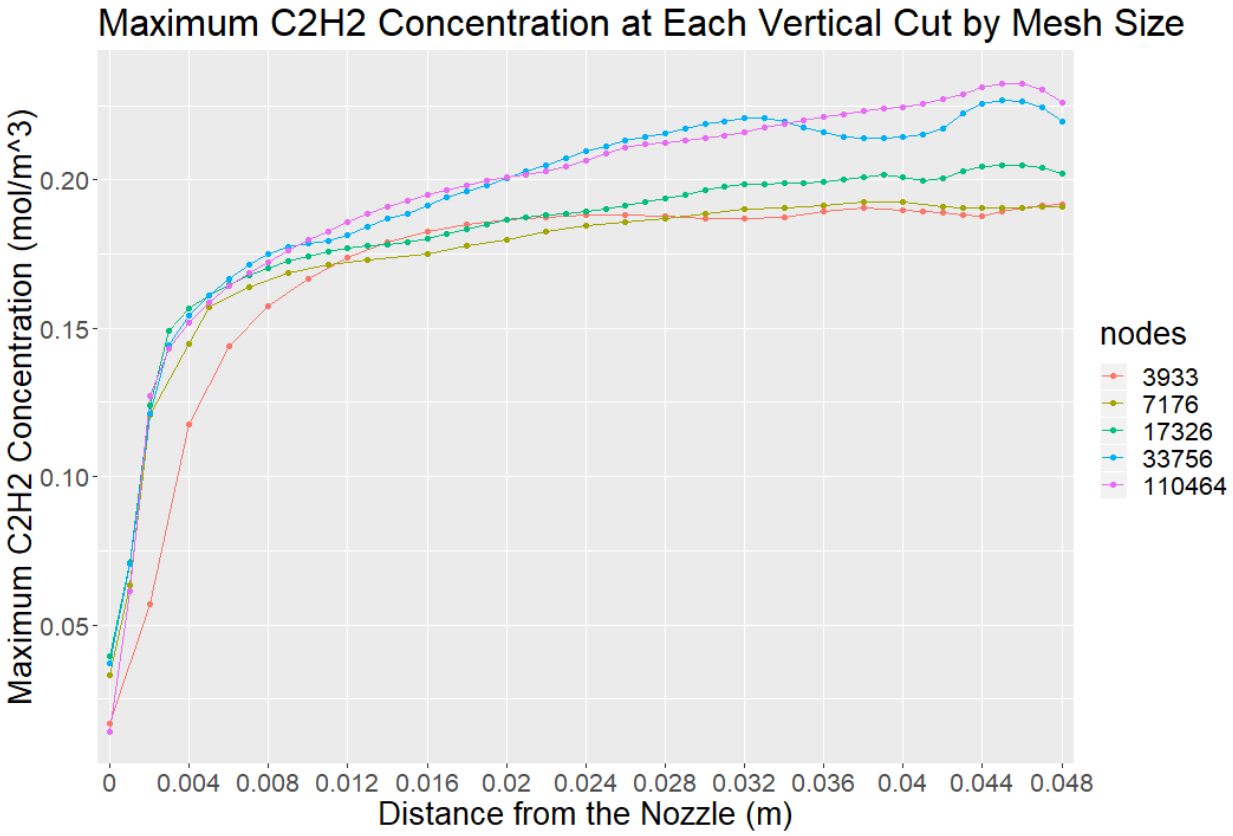

Figure 29: C2H2 concentration at each HAB

These results are more mesh independent, similar to the other properties observed. Thus, one can conclude that the C6H6 concentration is the most sensitive quantity in the solution.

4.2 Processing Chemkin Solution for 5-Step Mechanism

Once the initialization method was checked to ensure it still yielded the same solutions as those without any imported initialization files, the data could be processed to get insights into the time it takes for the flow to develop, the maximum temperature at each vertical coordinate, the half width temperature at each vertical component, and the mixing layer thickness at each vertical coordinate. The solution data was exported from Fluent into RStudio to conduct this analysis.

Fluent grouped the data based on the node number in the mesh. There were 20,493 nodes, so there were 20,493 data points. 20 variables were pulled out of Fluent, including node number, coordinates, directional velocities, temperature, pressure, species concentrations, density, molecular viscosity, and heat of reaction. This was all required to perform the necessary mutations.

The data was grouped based on the value of the vertical coordinate and took the maximum and minimum temperature for each distance from the nozzle. The maximum temperature figures were saved and the half width temperatures were solved for using the following equation:

$$T_{HW} = T_{min} + \frac{T_{max} - T_{min}}{2} \quad (7)$$

Where T_{HW} is the half width temperature, T_{min} is the minimum temperature, and T_{max} is the maximum temperature. Once these values were calculated, one could find the locations where these values were realized in the simulation. The data frame with the maximum temperature and half width temperature was merged with the main data frame. A script was written that went through each location and flagged the point if the temperature at that point was the maximum or half width temperature. For most profiles there were no points that were exactly the half width temperature as defined. To get a general understanding of where in the mesh the half width temperature would occur, seven percent error was allowed between the values from Eqn. 7 and the results of the simulation. This amount of acceptable error was the minimum threshold to obtain at least two half width locations at each vertical cut in the simulation. These locations were shown on the grid in Fig. 30 and Fig. 31.

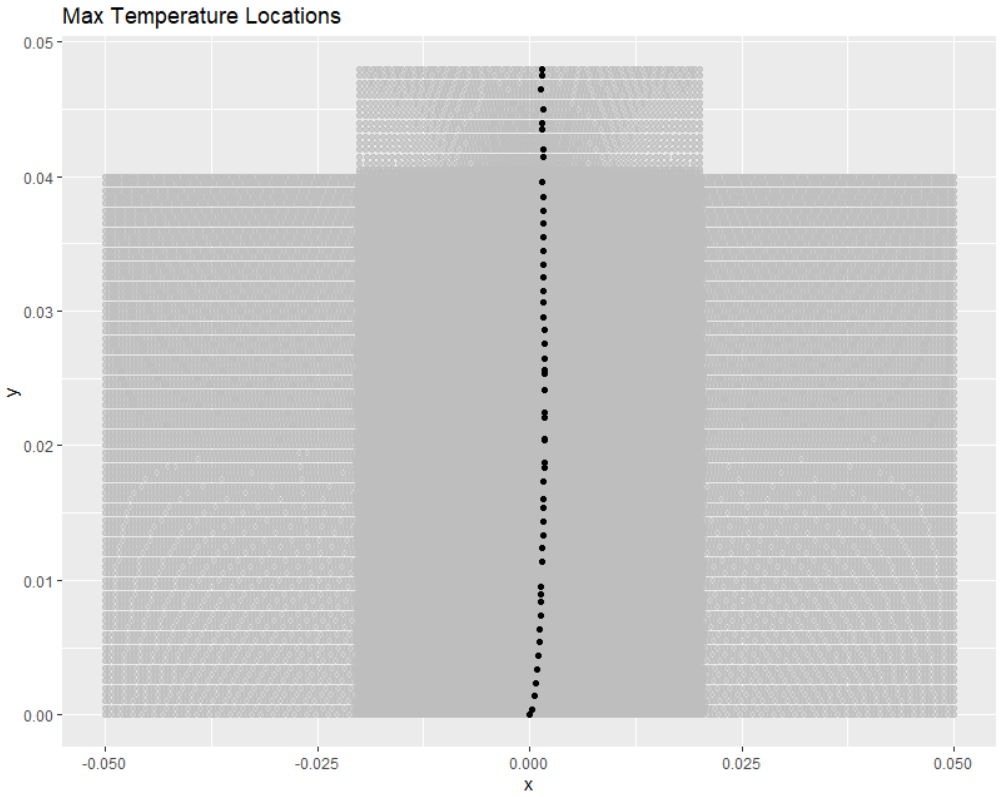

Figure 30: Maximum temperature locations

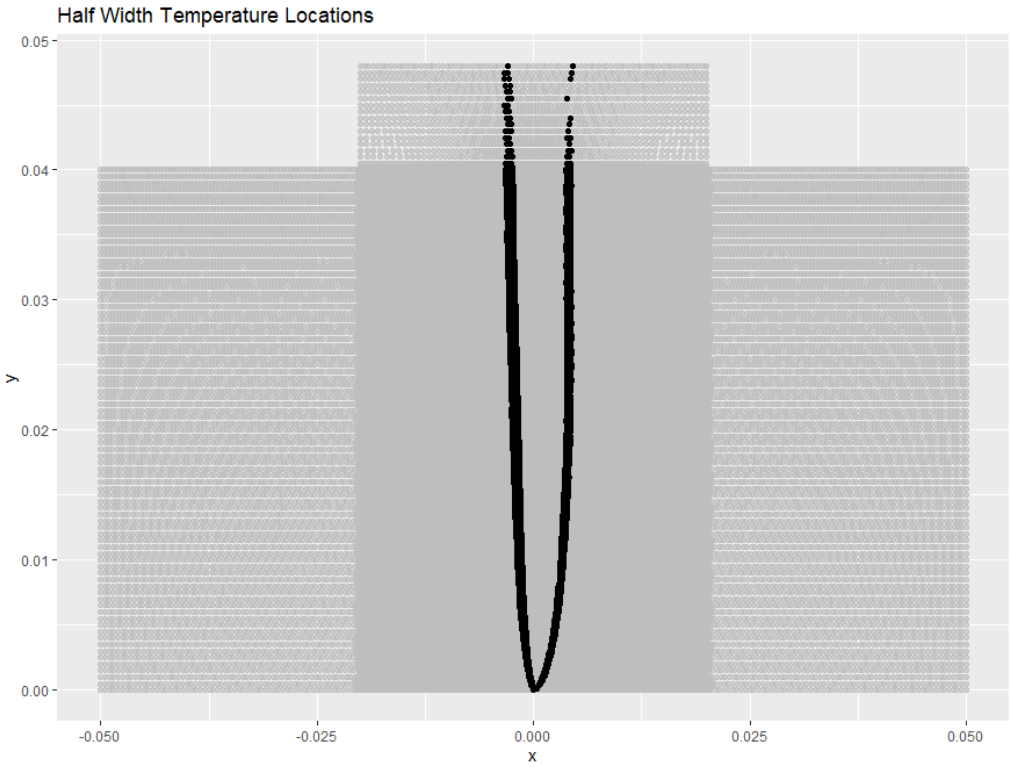

Figure 31: Half-width temperature locations

The temperature profiles were at each vertical coordinate to check that the simulation yielded sensible results. If the solution was not converged, one would expect those profiles to not be smooth. These profiles are also of use in that they provide a visual representation of the continuously changing horizontal location of the maximum temperature and the change in thickness of the mixing layer. As the vertical coordinate increases, the top of the curve shifts to the right and the distance between each half-way point of the curve increases. The horizontal distance between the half width temperature locations for a given distance from the nozzle is the mixing layer thickness.

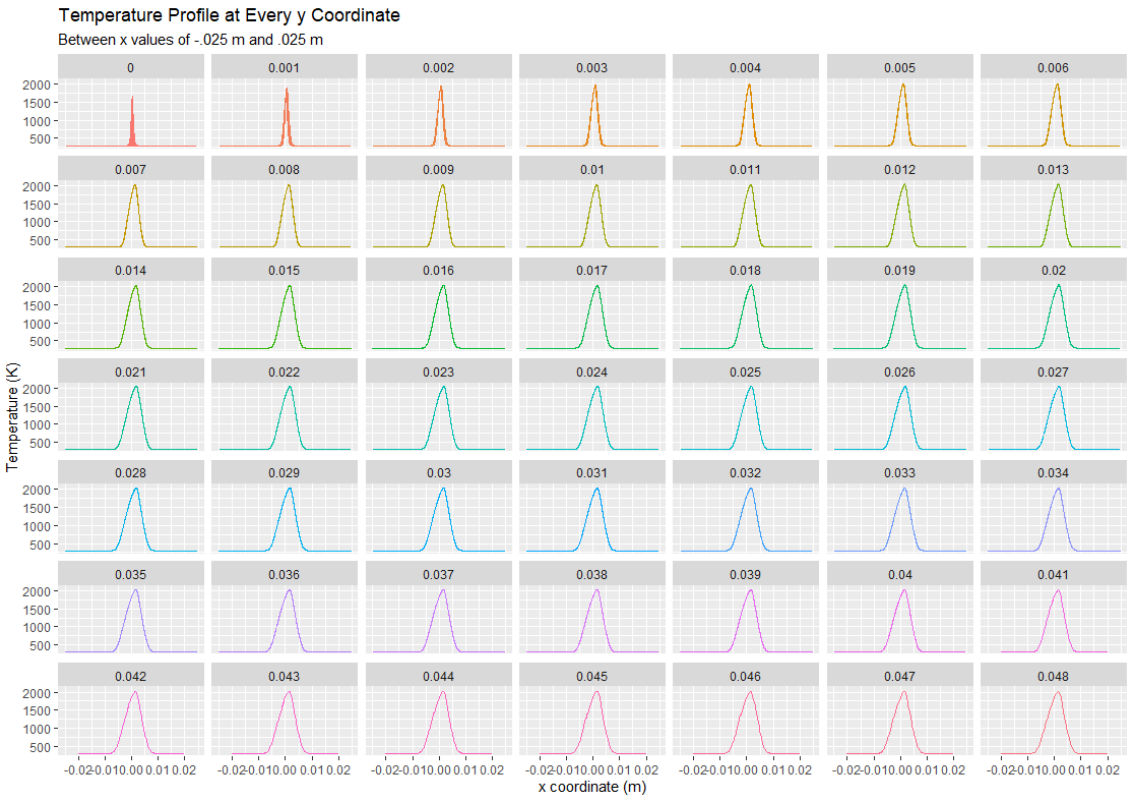

Figure 32: Temperature profiles for USC at each HAB

4.2.1 Mixing Layer Thickness for 5-Step Mechanism

With the data at one's disposal, the mixing layer thickness could be calculated with two distinct methods: the first computational, the second based on the solution data processed. The first method required a formula:

$$\delta = \sqrt{t * \frac{\mu}{\rho}} \tag{7}$$

Where δ is the mixing layer thickness, t is the time, μ is the molecular viscosity, and ρ is the density. The time was calculated by isolating the data points with the maximum temperature of each vertical cut. The change in time between each points was the change in vertical position from the previous point to the given point divided by the average of the velocity between those two points. By ordering the data frame by the value of the vertical coordinate, the changes in time could be added up time and the cumulative sum would represent the total time. The viscosity and density were variables exported from Fluent. The change in the estimated mixing layer thickness over time is depicted in Fig. 33.

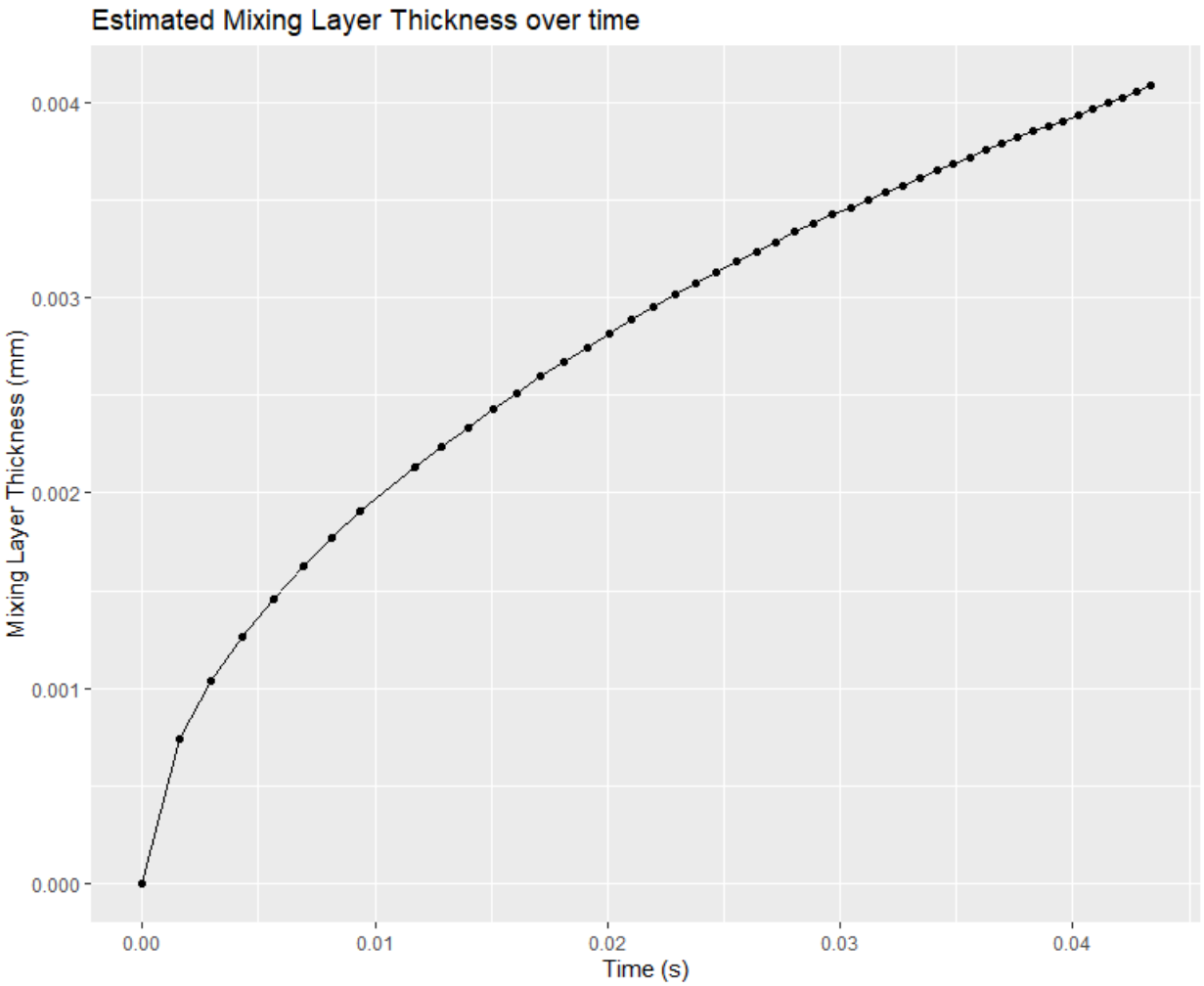

Figure 33: Estimated mixing layer thickness

To calculate the experimental values explained above, the vertical coordinates with multiple points registered had to be first stripped out as half width temperature locations based on my methodology. This was remedied by stripping out coordinates with values that were the same within three significant figures. The data was then parsed and the code was written to find the difference in horizontal position for any pairs of points that had the same vertical location. Furthermore repetitive points were stripped by taking the mean of the mixing layer thickness for

vertical coordinates that were the same within three significant figures. These values were joined to the main data frame using a for loop. An empty list was created and then the main data frame was looped through in sections with the same vertical location. The mixing layer thickness values were appended to the smaller data frame and stored as a section in the empty list. After all the vertical coordinates were accounted for, each chunk was picked out and placed back together into one data frame. Once the locations and approximate distances located between half width temperature points were identified, the points were interpolated so that the shape of the experimental mixing layer thickness values with respect to time met the expectations established by prior research into fuel and oxidizer mixture flows. The visualization in Fig. 34 shows experimental mixing layer thickness calculations based on the temperature values of each point in the grid and how those compare to the computational calculations explained before.

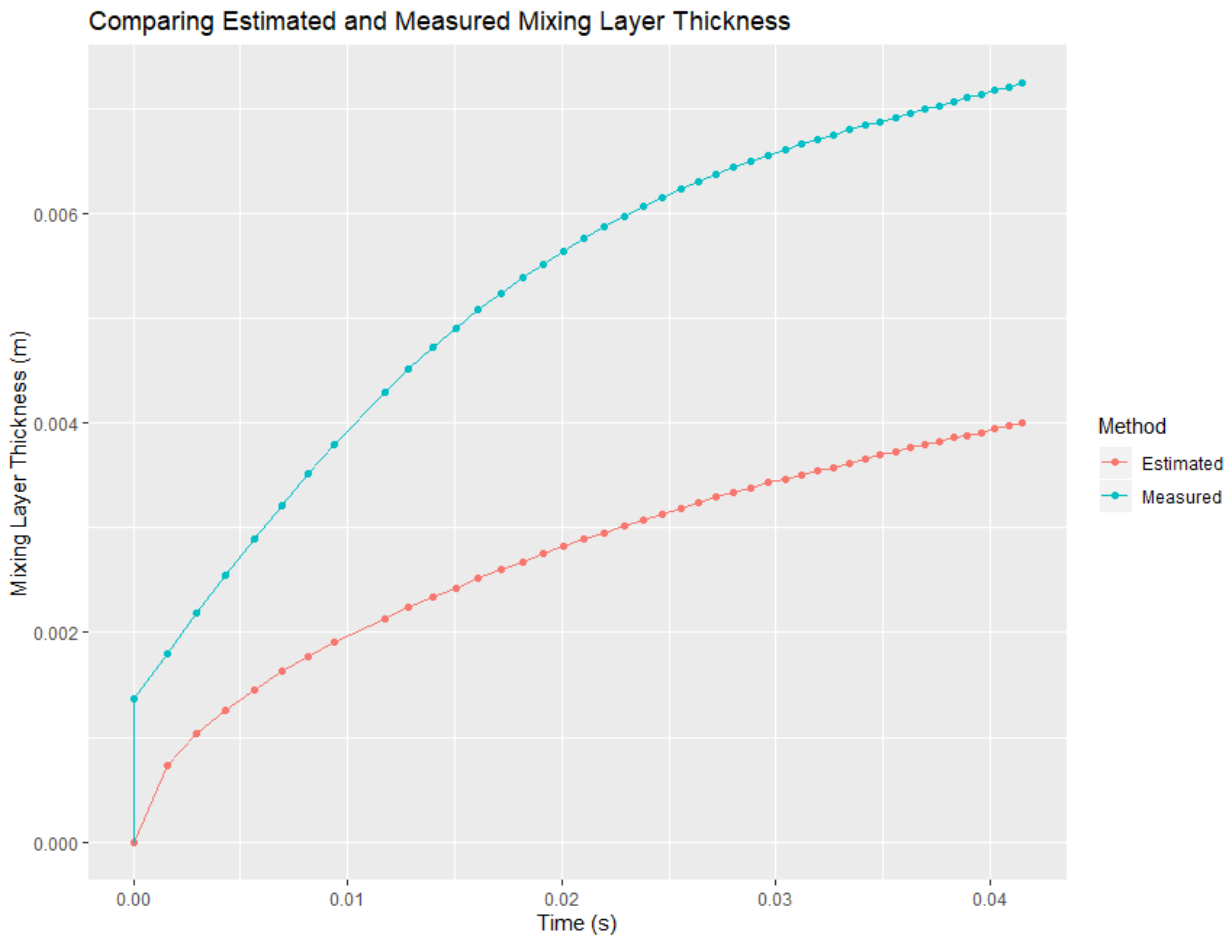

Figure 34: Comparison of the measured and estimated mixing layer thicknesses

As one can see, the values yielded from the experimental procedure are consistently larger than those from the computational method. This was attributed this to the environment of the simulation, which is not accounted for by the model demonstrated in Equation (2). The rate at which the thicknesses increase in each case is relatively similar, which manifests itself in a close to linear relationship, per Fig. 35.

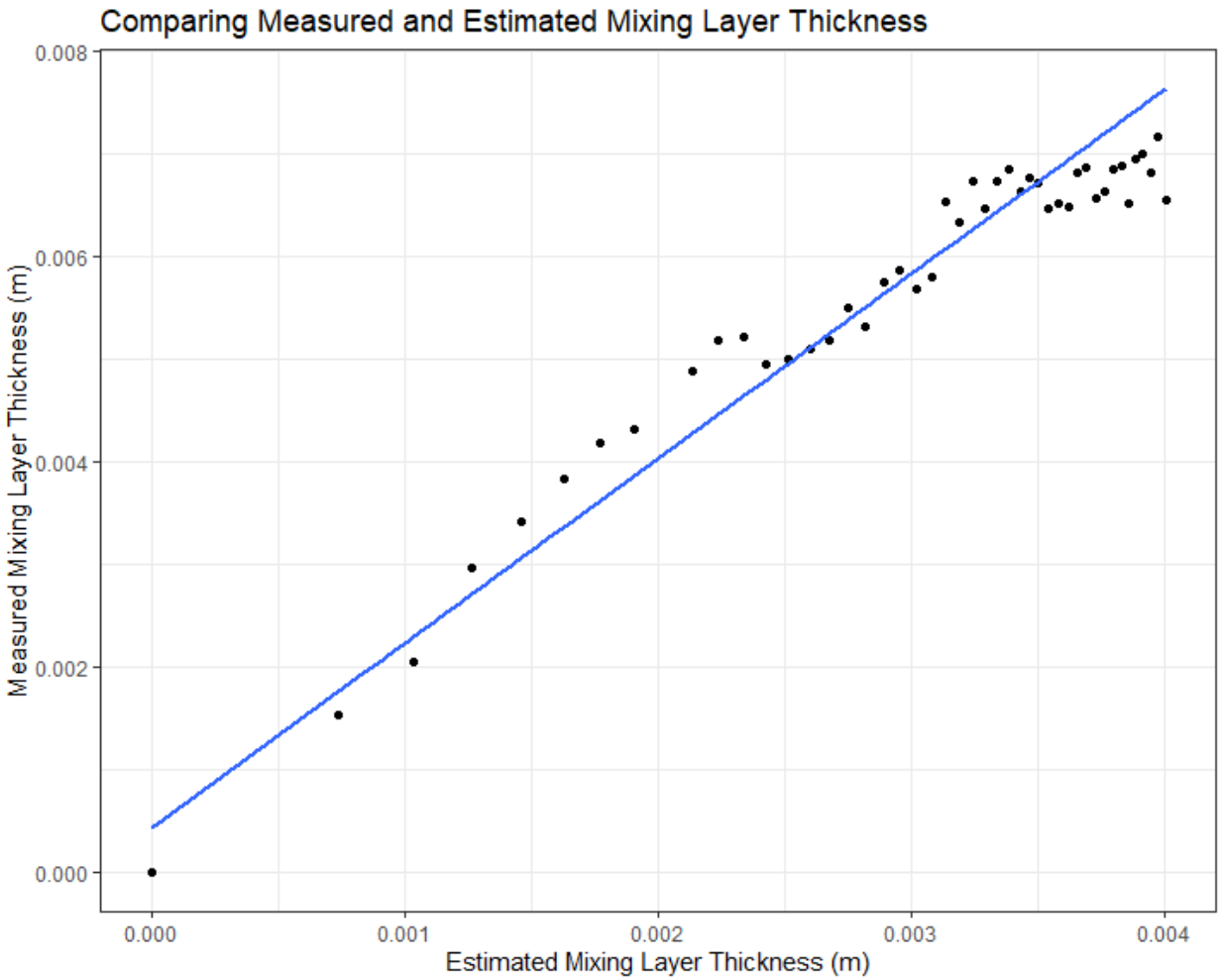

Figure 35: Direct comparison between mixing layer calculations

The main purpose of solving for the mixing layer thickness with the simulation data was to normalize the temperature profiles and check that when accounting for the maximum temperature at each vertical location and the mixing layer thickness, the temperature profiles should be the same throughout the entire system (with the exception of when time equals zero). The normalized temperature is the ratio of the given temperature for a point and the maximum temperature at the point's profile. The normalized horizontal coordinate is the ratio of the difference in the

given coordinate and the coordinate of the maximum temperature to the experimental mixing layer thickness. These profiles in Fig. 36 are the same set as above, except with the ordinate and abscissa normalized as described.

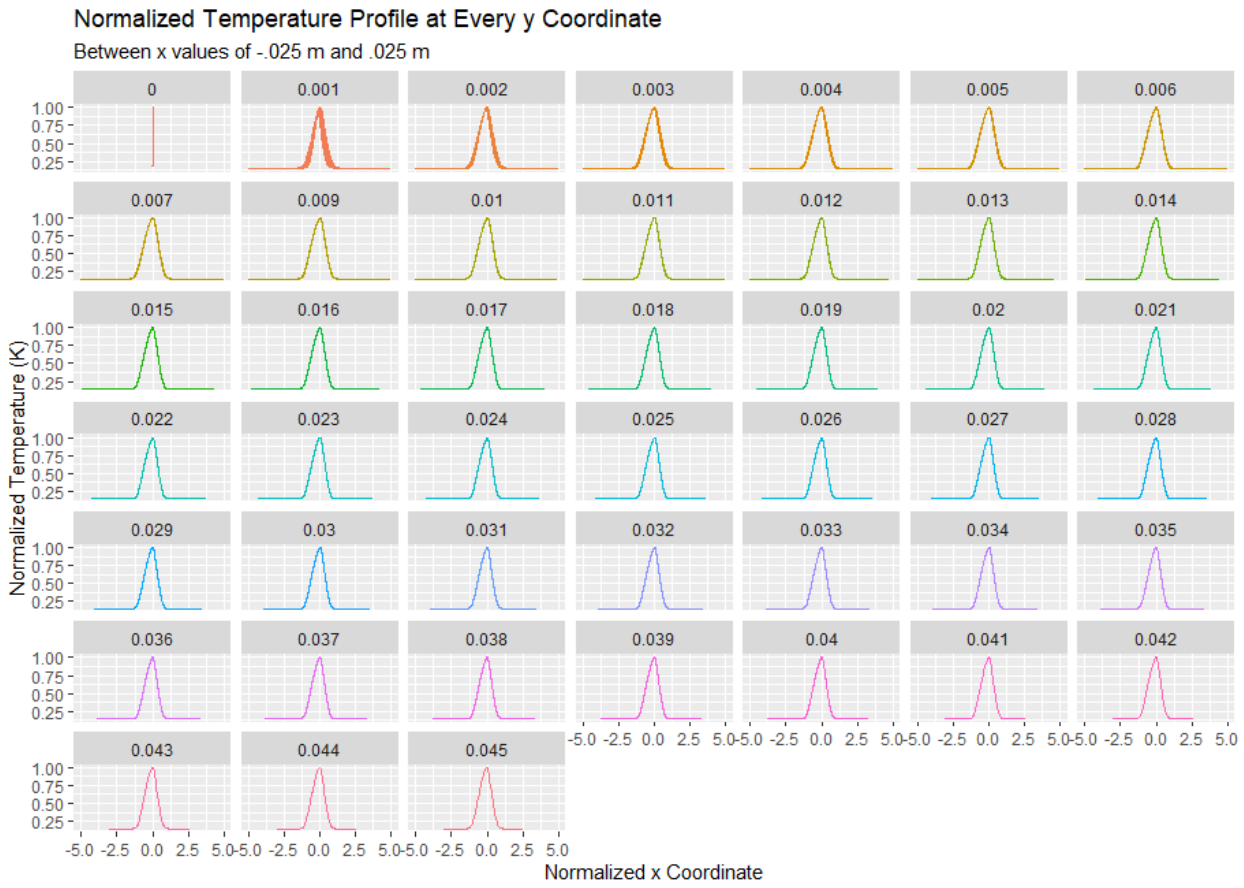

Figure 36: Normalized temperature profiles for 5-Step mechanism

The profiles look are the same throughout the mesh. The maximum temperature always occurs at the normalized horizontal coordinate of zero because the coordinate is always subtracted by the location of the maximum temperature. The thickness of the profiles are the same because each coordinate is divided by the mixing layer thickness at each point. And since the temperature is divided by the

maximum temperature at a given vertical cut, the height of each profile is the same. These phenomena might not be obvious from looking at each profile, so in Fig. 37 is the same data filtered by just four different coordinates with all of the profiles on one plot.

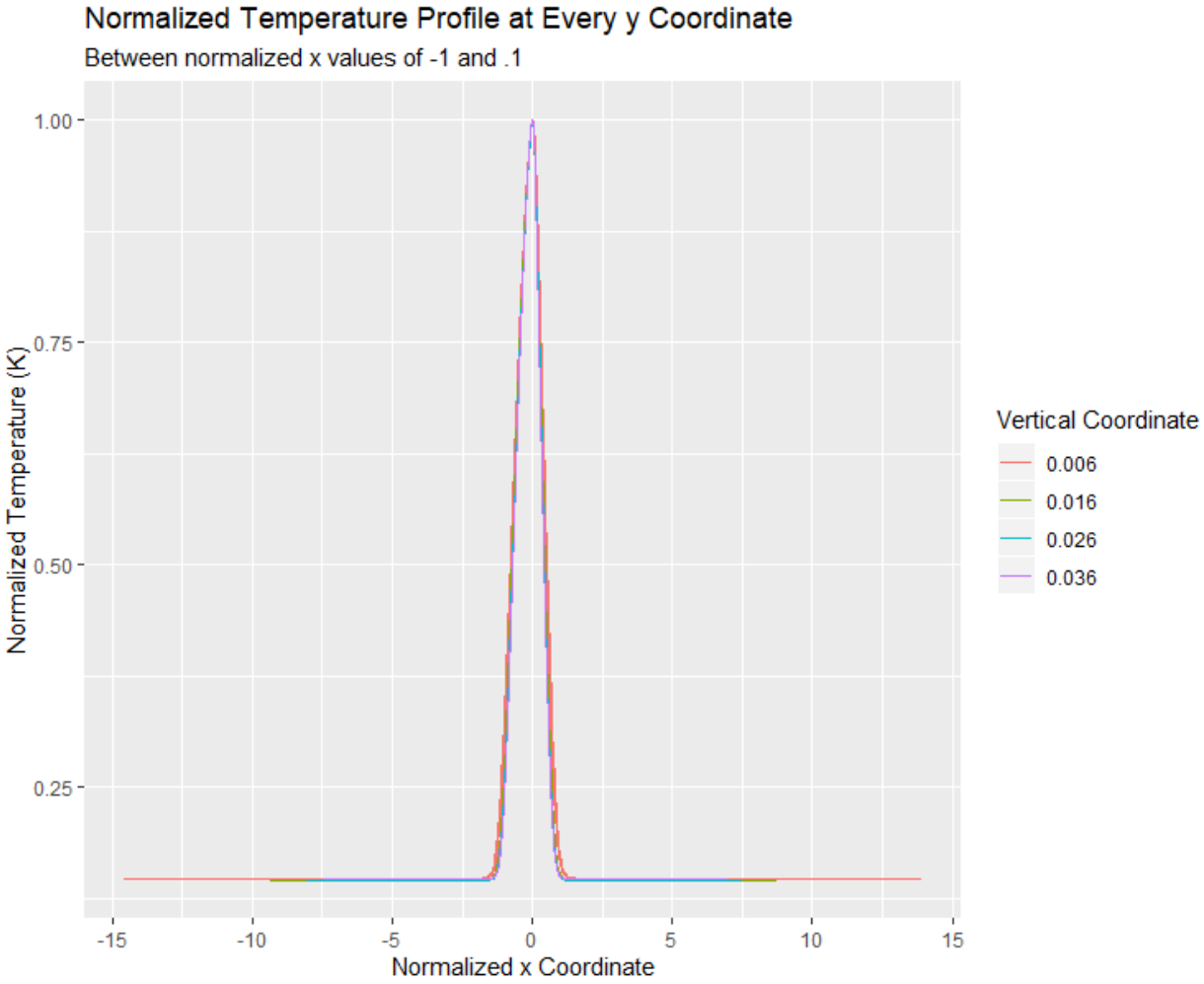

Figure 37: Normalized temperature profiles on one panel for 5-Step

The solution data acted as expected. This gives confidence in both the solutions results and the data processing methods. From here, one can apply the same

techniques to the more complicated USC2 system with 110 species in the environment.

4.3 Processing Chemkin Solution for USC2 Mechanism

The horizontal temperature profiles computed with the USC mechanism are presented in Fig. 38 and show the development of the mixing layer. As the flow moves downstream from the inlets, one can observe the thickening of the profile. This indicates that the region in which the fuel and oxidizing agent are interacting is growing with the generation of heat at steady state being balanced by diffusive losses. The interaction of the two flows yields results in the occurrence of exothermic reactions and in the establishment of a non-uniform temperature. The maximum temperature at low HABs is much smaller than those at higher HABs because the conductive heat losses are the most significant when the mxing layer is the thinnest.

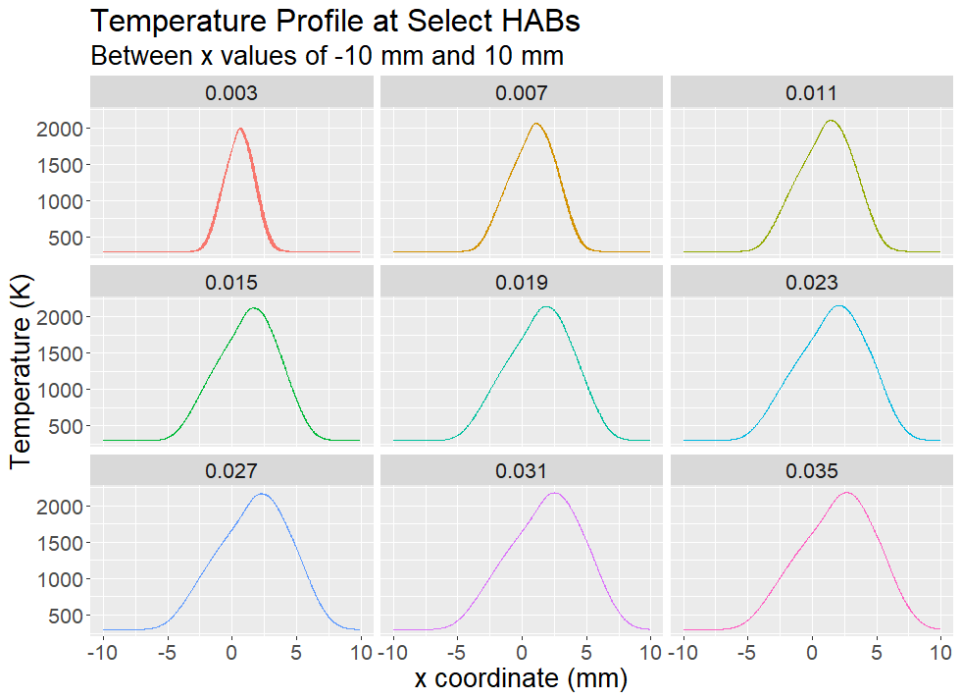

Figure 38: Temperature profiles at select HABs for USC mechanism

The profiles of other relevant quantities were observed to affect the mixing layer thickness as well, depicted in Fig. 39 to 41. The profile of the horizontal velocity component in Fig. 39 and of the concentration profiles of relevant species in Fig. 40 and 41 do not show that same shape as the temperature. However, the same balance between production and diffusion losses is at play for the species.

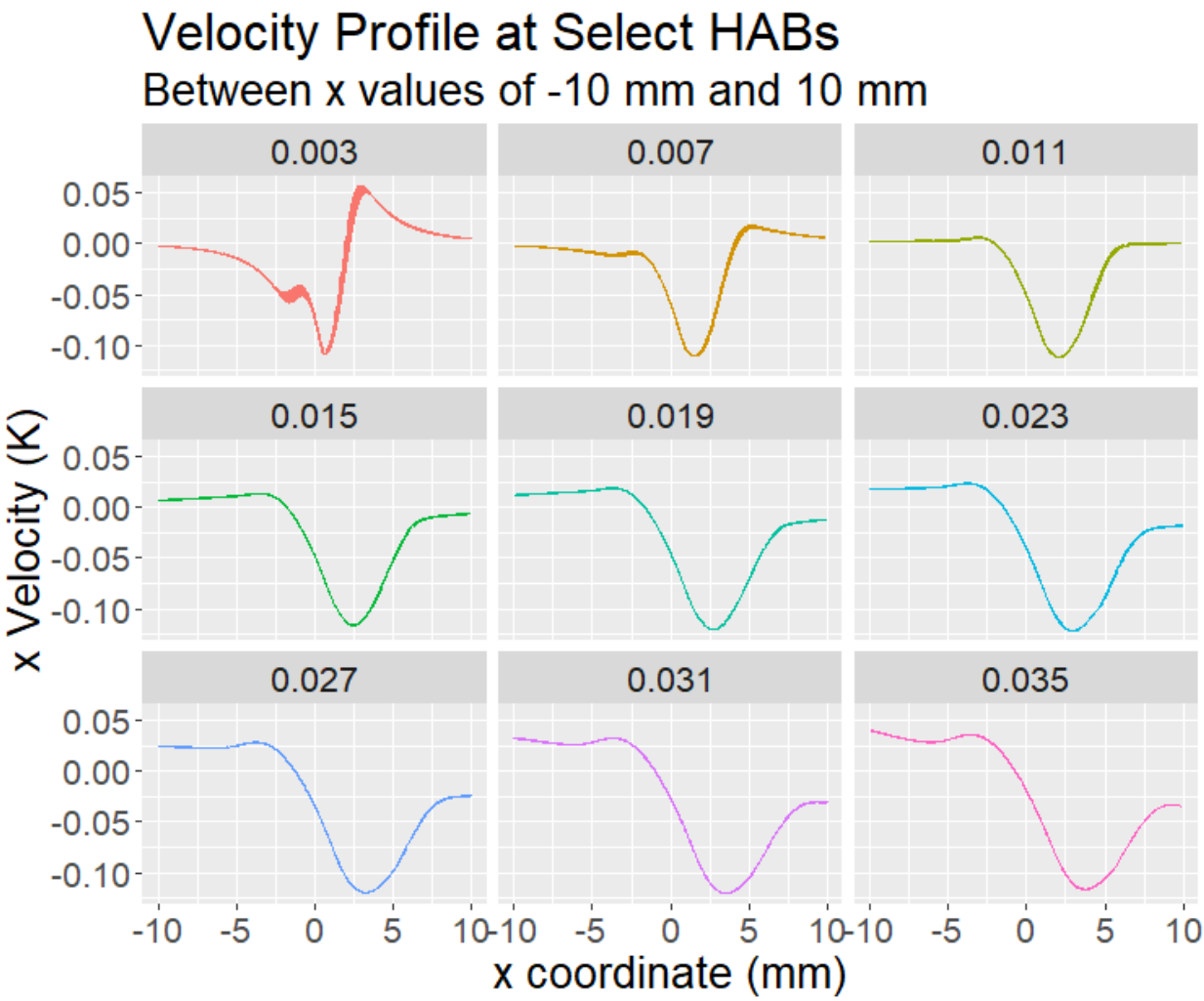

Figure 39: Horizontal velocity profiles at select HABs for USC mechanism

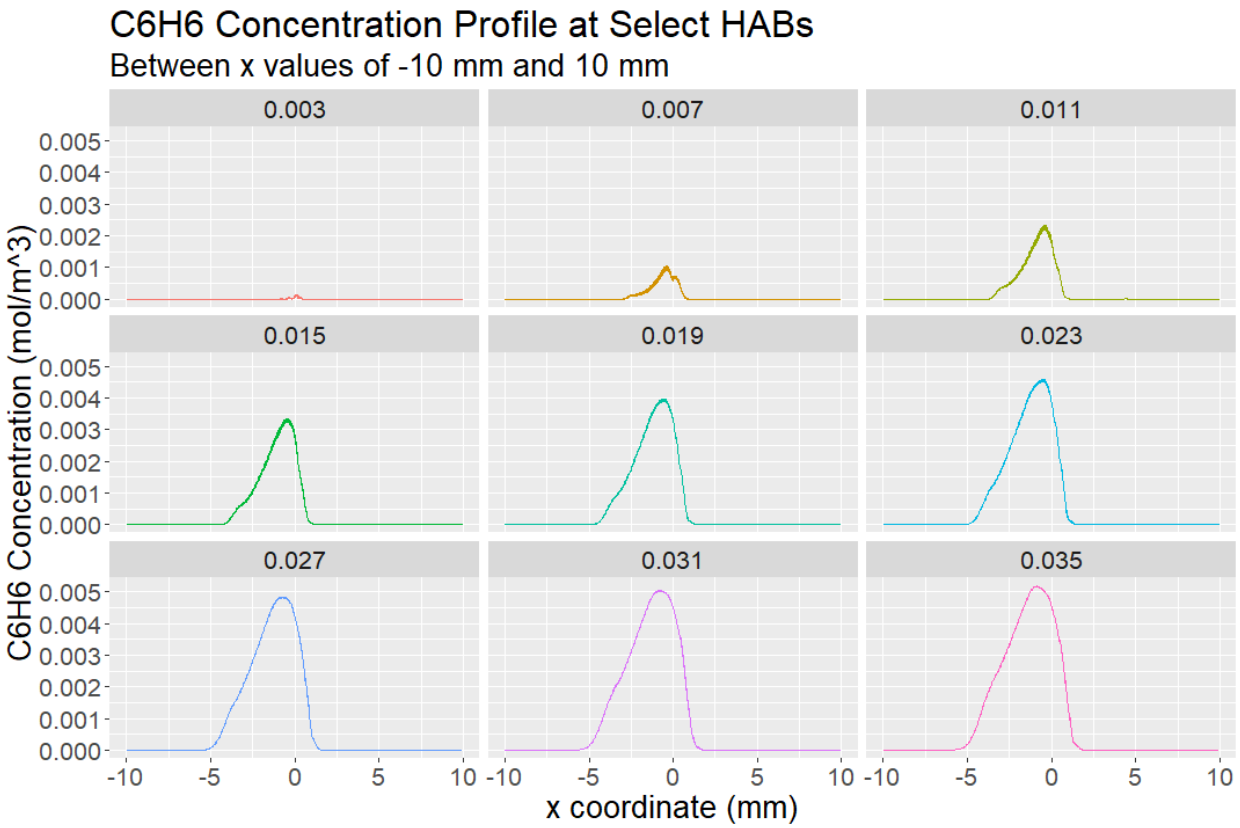

Figure 40: C6H6 profiles at select HABs for USC

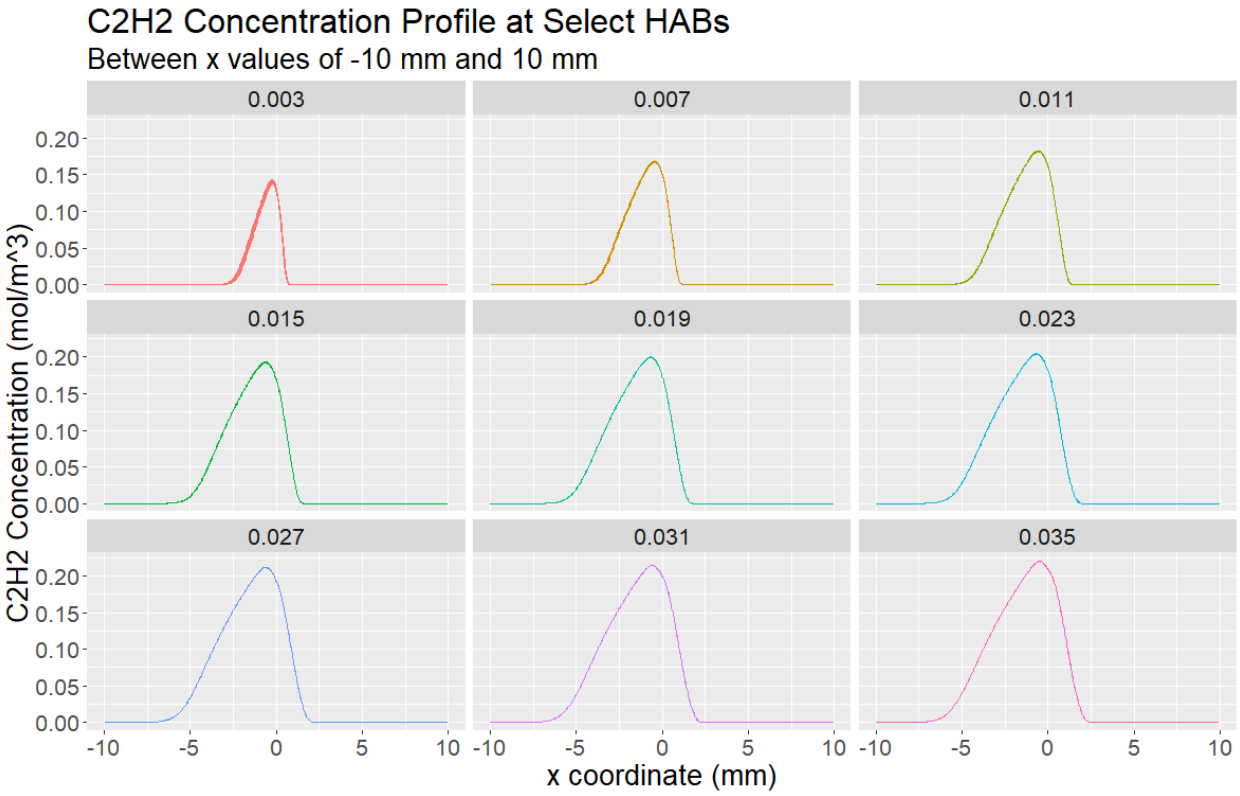

Figure 41: C2H2 profiles at select HABs for USC

Given that the mixing layer thickness and maximum temperature increase with time, the profiles in Fig. 38 through 41 at different HABs are different. However, if the profiles are normalized based on the maximum temperature and the mixing layer thickness, they should be similar for each vertical coordinate.

The solution data was processed to locate the maximum temperature of each vertical coordinate in the USC2 mechanism, as depicted in Fig. 42. These locations were essential for this study. They were required to calculate both the estimated and measured mixing layer thicknesses.

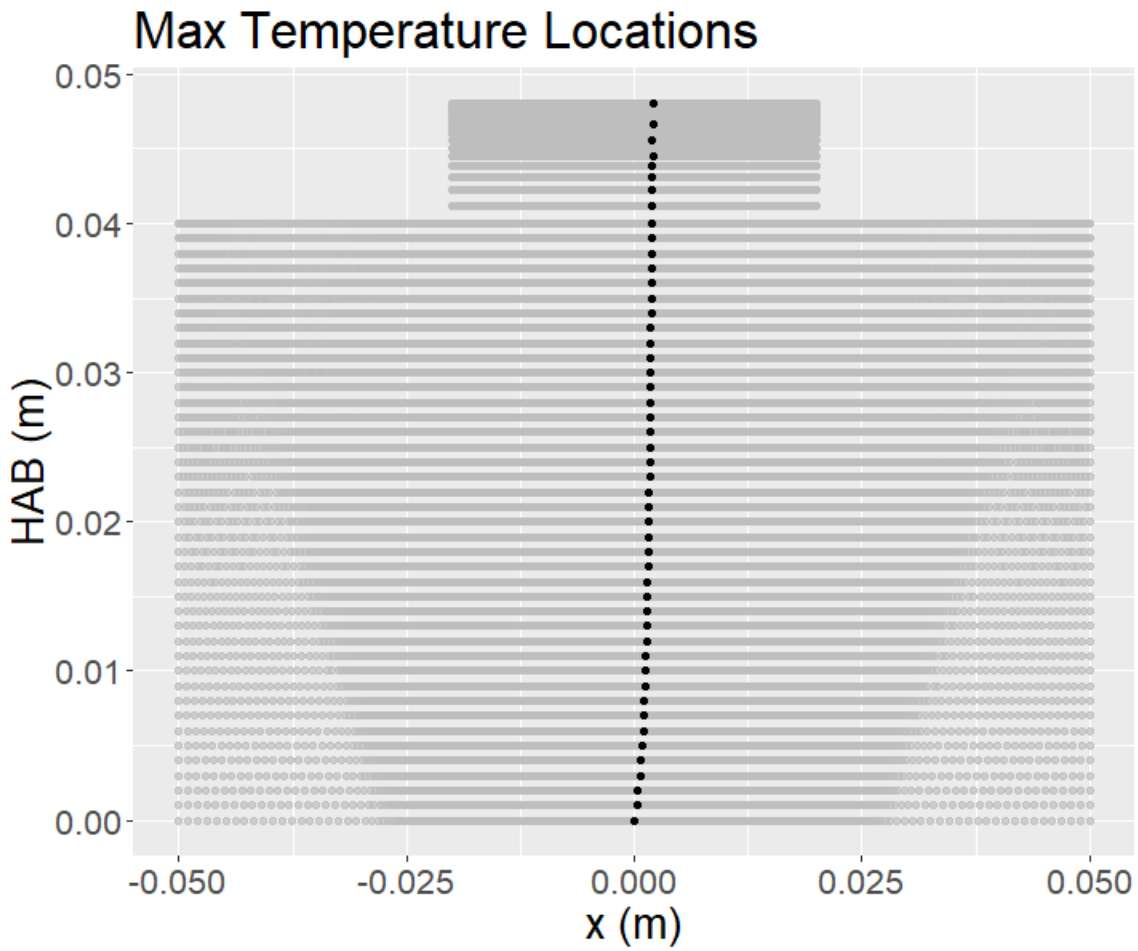

Figure 42: Maximum temperature locations for USC mechanism

4.3.1 Mixing Layer Thickness for USC2 Mechanism

The convective time for the flow to move downstream needs to be calculated, if one wants to estimate the thickness of the mixing layer since the latter grows with time. Using the methods described in section three, the change in time between points on the maximum temperature line were calculated based on knowledge of the vertical velocity and displacement between two points, the

change in time between two points. The solution data from Fluent also include values of the density and molecular viscosity. These quantities yield the kinematic viscosity. The cumulative time at each point was the cumulative sum of the changes in time moving from the fuel to oxidizer nozzles at HAB = 0 mm. With the kinematic viscosity and time at each point in hand, one can calculate the estimated mixing layer thickness using Eqn. 8.

With regards to the measured mixing layer thickness, the maximum temperature locations also proved to be vital. The measured mixing layer thickness is defined by the distance between half width temperatures for a vertical coordinate, per section three. To get the half width temperatures at each vertical coordinate, the maximum and minimum temperatures for each location were required based on Eqn. 7.

The half width temperature locations are the basis for the measured mixing layer calculations, show in Fig. 43. Understanding how the location of the half width temperatures evolve is the crux of entire study. The benefit of using the PML configuration is contingent on achieving comparatively large thickness of the mixing layer compared to other flame configurations.

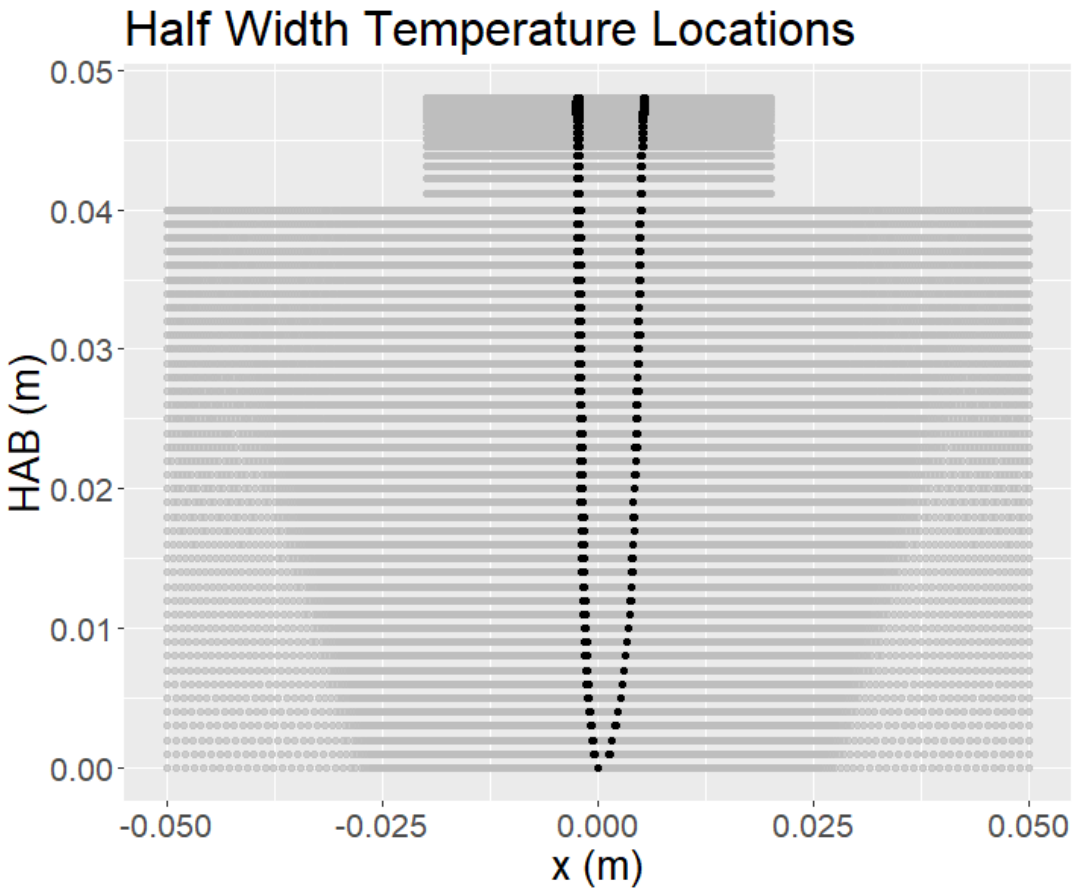

Figure 43: Half-width temperature locations for USC mechanism

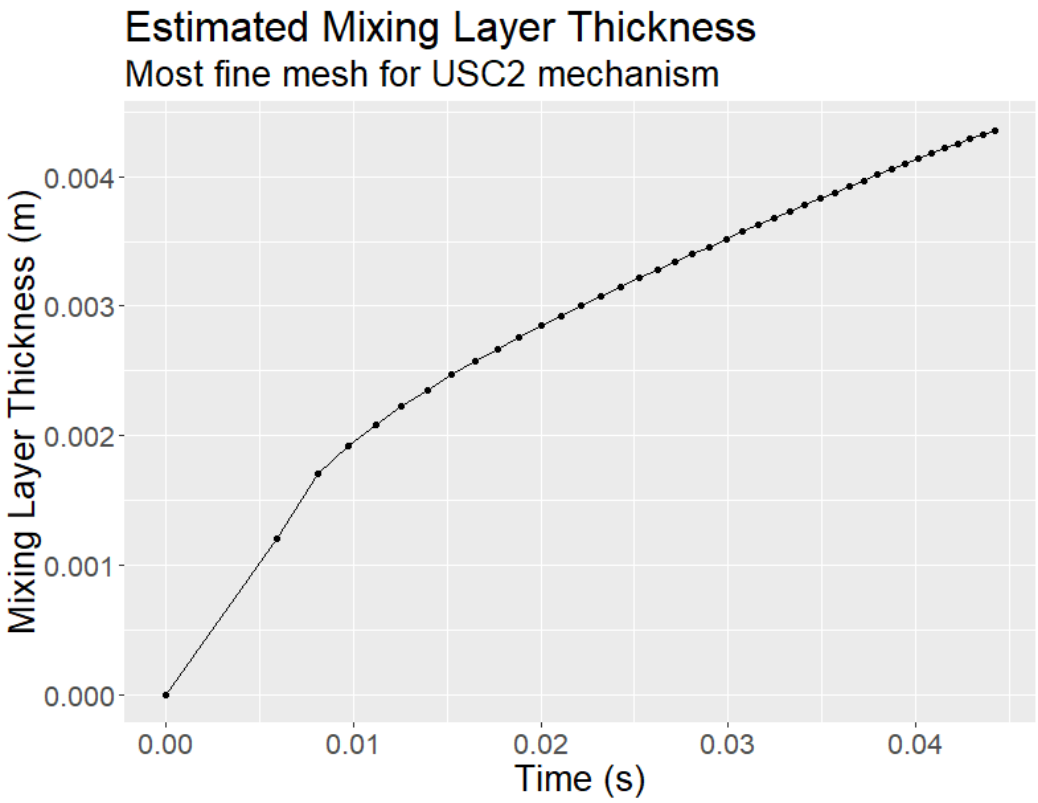

Figure 44: Estimated mixing layer thickness calculations for USC mechanism

The size of the mixing layer thickness increases over time, consistent with the observations from the non-normalized temperature profiles. The rate at which the size increases, however, decreases with time. It increases monotonically, but the rate is not linear nor exponential. Instead, it takes the shape of a root function. There are diminishing returns for the change in mixing layer thickness as HAB grows.

Once the shape of the estimated mixing layer growth was established, the measured mixing layer growth can be analyzed. As shown in Fig. 45, the shape of the two curves are similar.

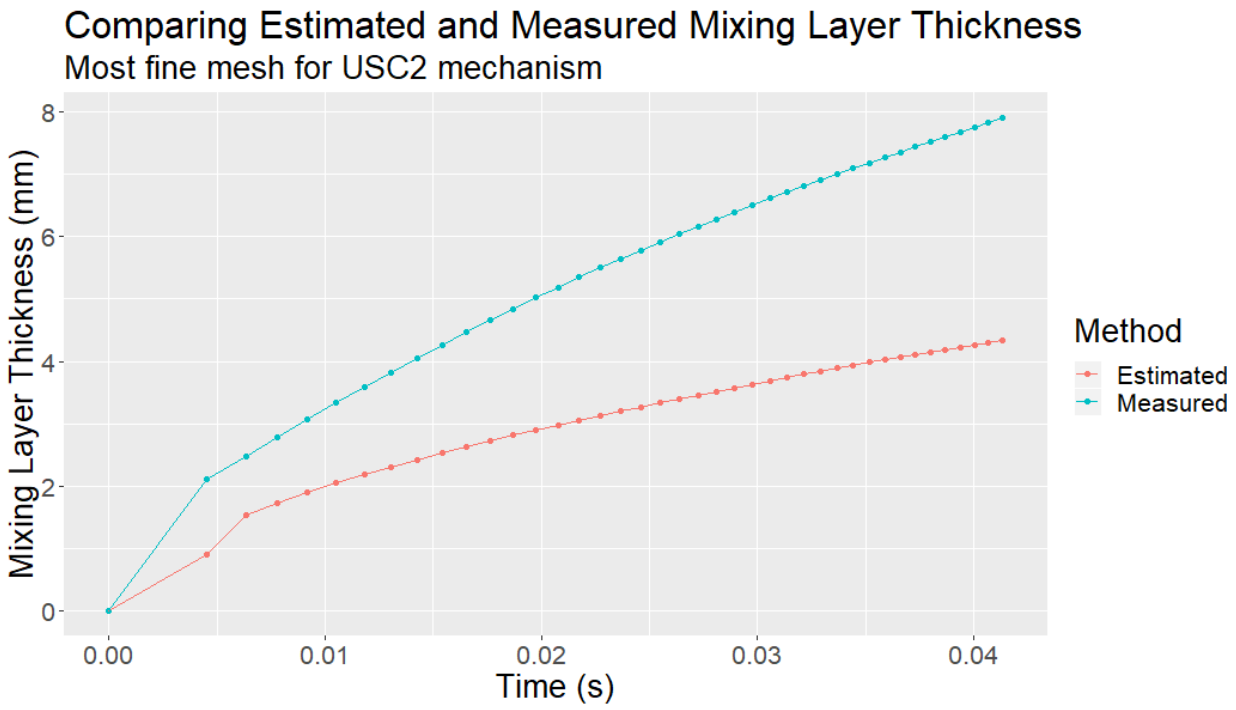

Figure 45: Both mixing layer calculations for USC mechanism

The actual values for the mixing layer thickness differ between the two methods because, due to the gravitational buoyancy, the vertical velocity is largest at the location of the maximum temperature where the residence time was estimated. At any rate, the fact that the growth profiles match should distill confidence in the methods for calculating the measured mixing layer thickness. Since the estimated mixing layer thickness has no basis in the actual location of the edges of the mixing layer, the estimation method should not be used to define the expectations for an actual physical experiment, unless a robust correlation can be developed. The measured mixing layer thickness is more relevant in this regard. The estimated values provide a convenient baseline for relative mixing layer

growth. That both methods undergo the same growth pattern should give any onlooker confidence in the figures derived when determining the feasibility of this configuration.

To further corroborate the idea that the two methods for mixing layer size calculations experience similar growth, the mixing layer thicknesses in each were directly compared. If the patterns are alike, the relationship between the two methods should be linear. Sure enough this is the case. The magnitudes of the values from each method yield close two a perfect linear fit, the blue line in Fig. 46.

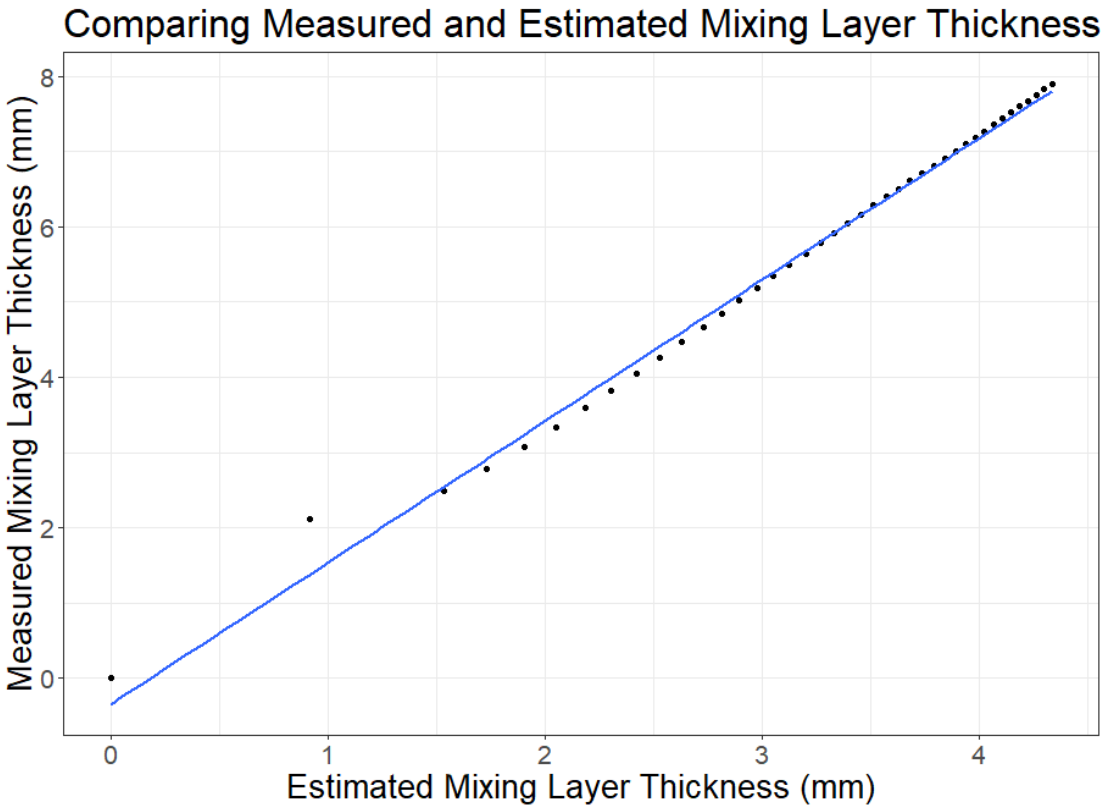

Figure 46: Comparison of mixing layer calculation methods

With this analysis in hand, the method used for explicitly computing the measured mixing layer thickness should be considered valid. It follows the implicit, theoretical model for mixing layer thickness while incorporating the nuances of the mechanism and configuration of the study. The theoretical model is not based on the system of this study. Thus, the actual values should not be taken accepted as physically realizable. The explicit estimations are unique to the USC2 mechanism and the PML configuration. The measured mixing layer thickness are the values of note when evaluating the merits of studying soot inception with this specific system.

The normalized temperature profiles were used to ensure that the measured mixing layer calculations were correct. Such normalized profiles are obtained over the maximum temperatures while the abscissa is scaled over the mixing layer thickness at the considered HAB.

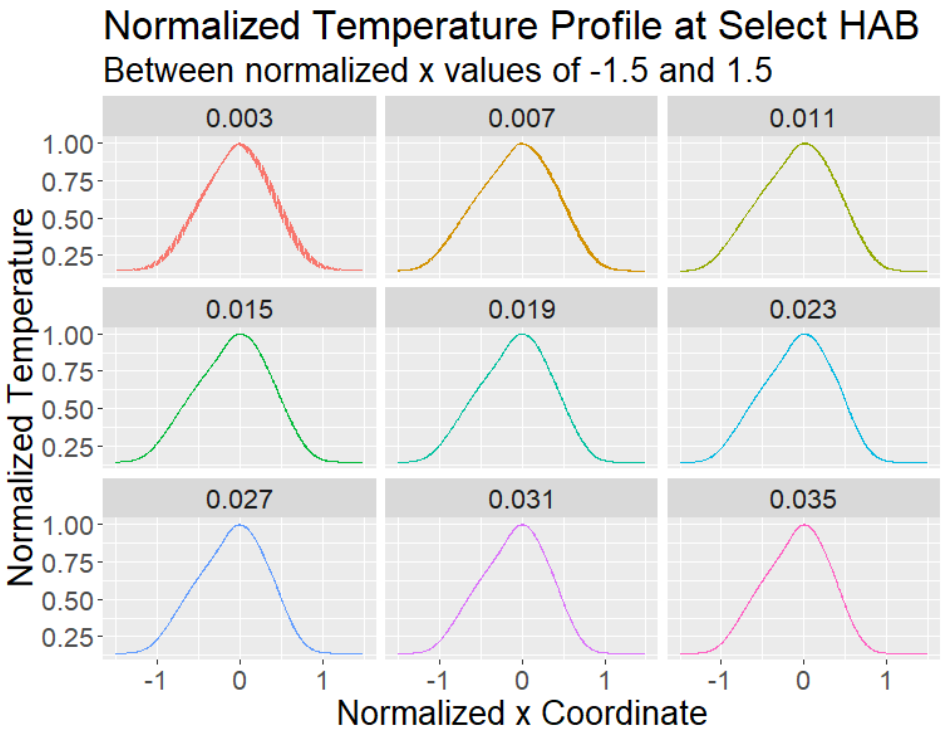

Figure 47: Normalized temperature profiles at select HABs for USC mechanism

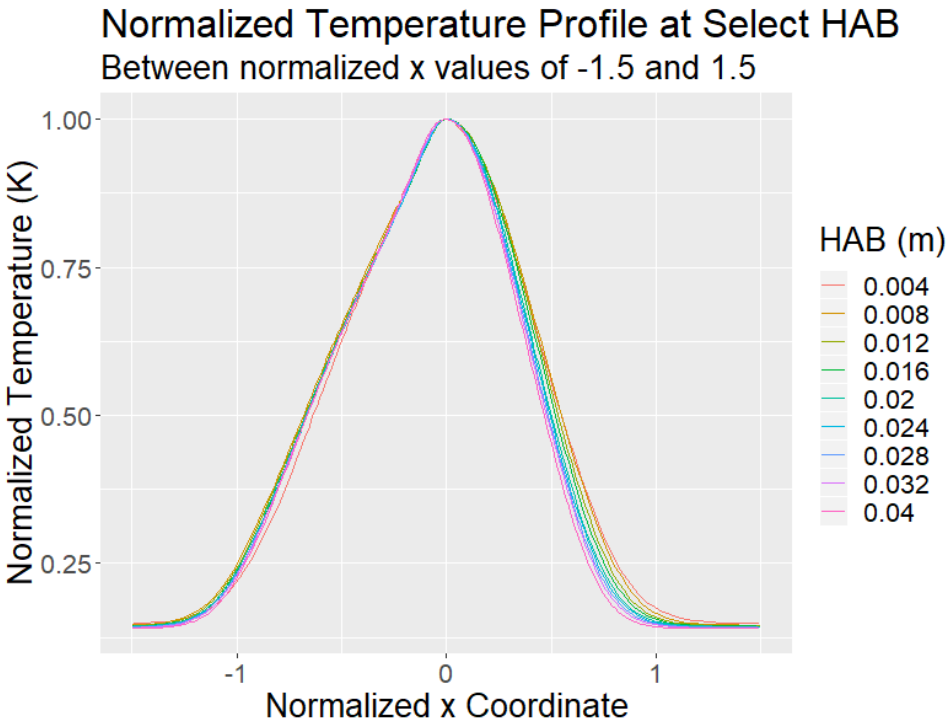

Figure 48: Normalized temperature profiles on one panel for USC mechanism

These profiles give credence to the fidelity of the calculations and confirm that conclusions based on the development of the mixing layer are useful when creating a physical experiment in the future.

4.3.2 Relative Locations of C6H6 and C2H2

Furthermore, the profiles for C6H6 and C2H2 were observed as were the corresponding normalized profiles in Fig. 49, Fig. 50, Fig. 51, and Fig. 52 respectively.

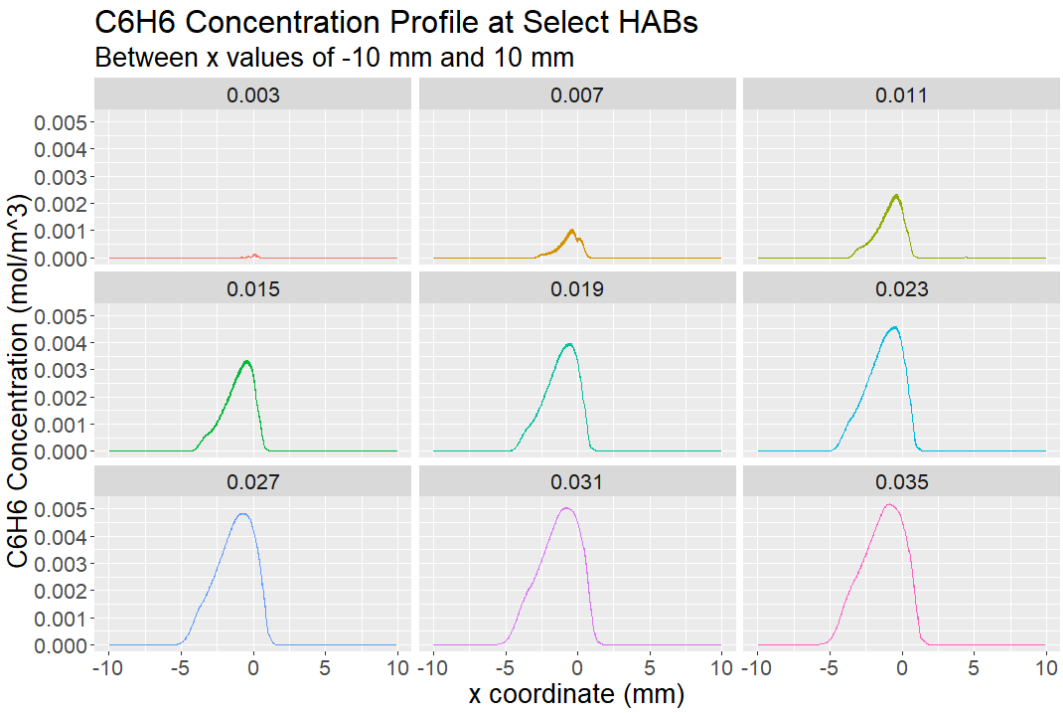

Figure 49: C6H6 concentration profiles at select HABs for USC

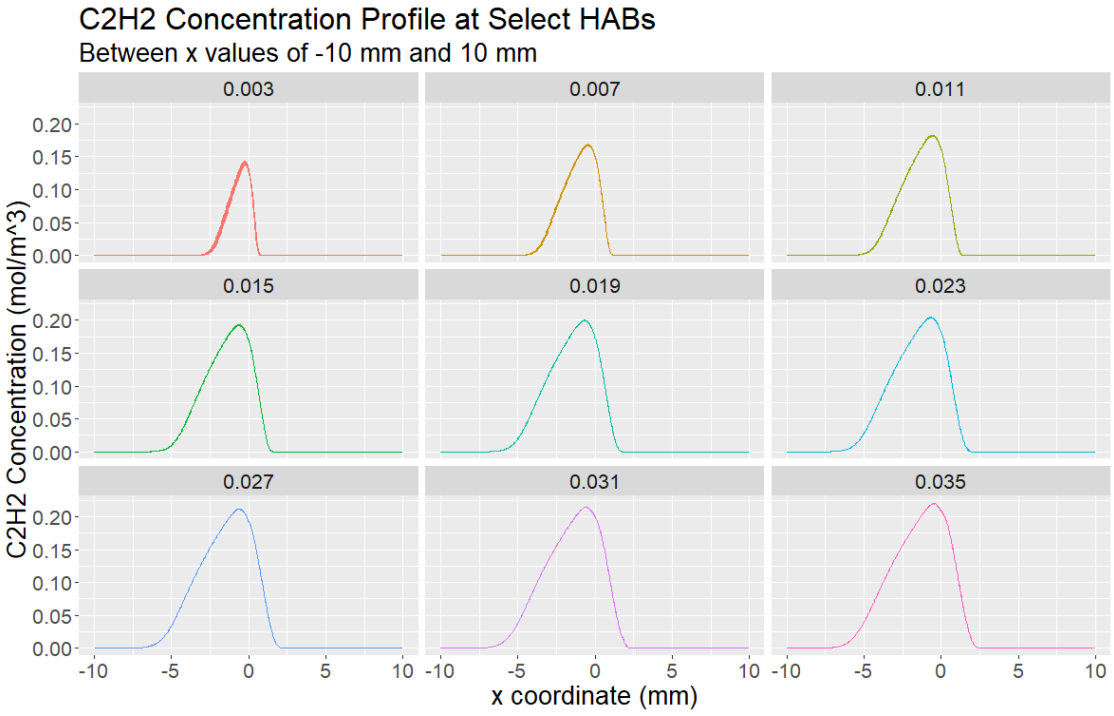

Figure 50: C2H2 concentration profiles at select HABs for USC

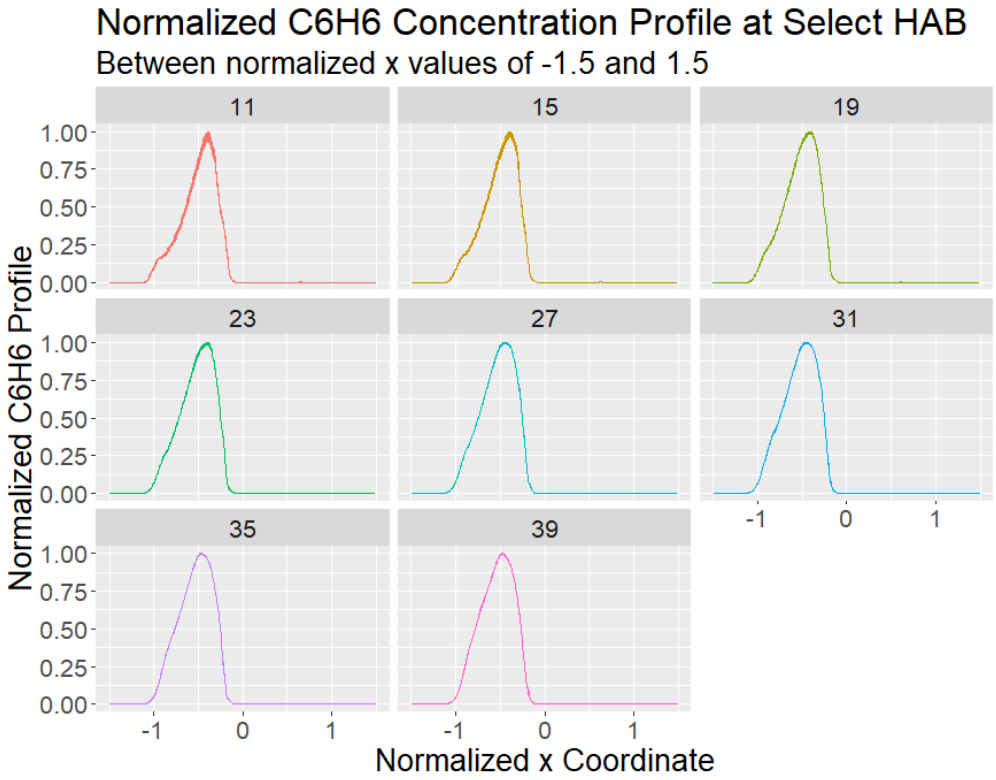

Figure 51: Normalized C6H6 profiles for select HABs for USC

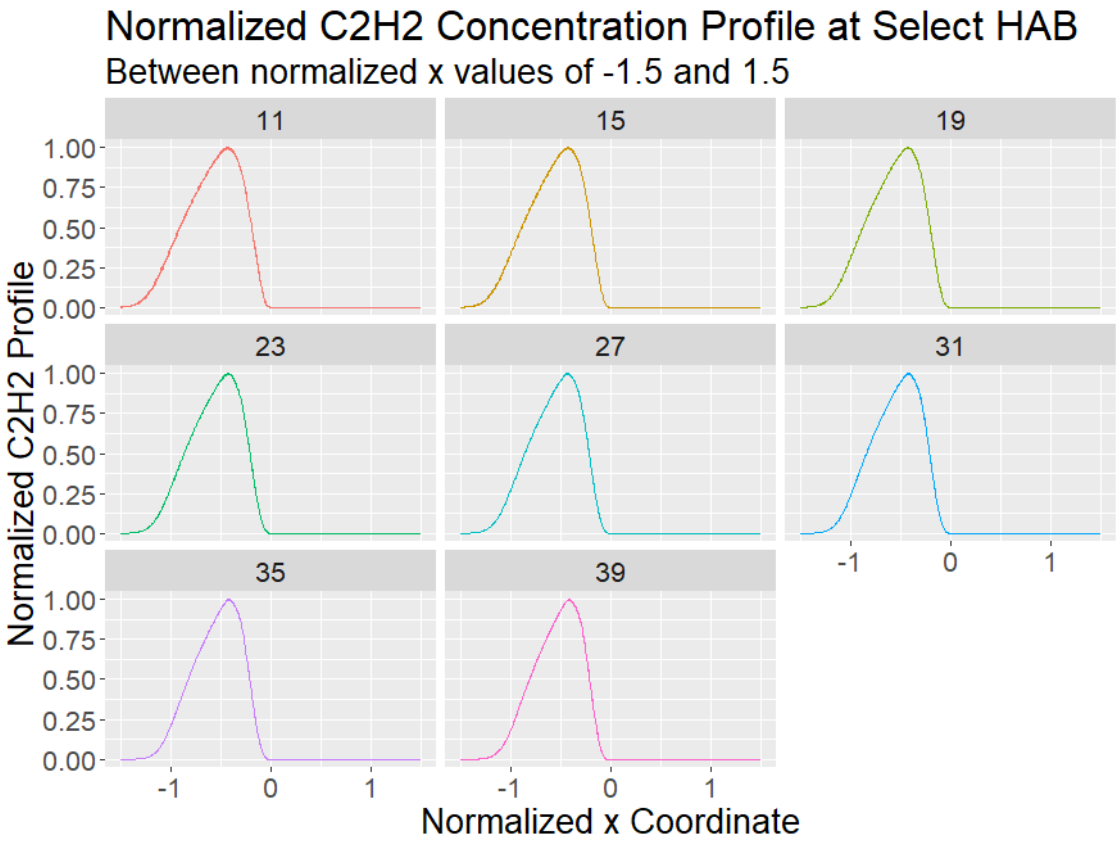

Figure 52: Normalized C2H2 profiles at select HABs for USC

For the normalized C6H6 profiles, one can see that the initial profiles do not look as smooth as the later profiles, especially at the earliest HAB of 11 mm. At even smaller HABs (i.e. less than 10 mm), the profiles look even less smooth with non-monotonic behavior before and after the peak concentration. This is attributed to the concentration of benzene not properly resolving early in the simulation. The mesh was adapted in such a way that there were more nodes at the splitter than at the inlets for the fuel and oxidizer jets. The concentration of benzene starts at zero (at the start of the simulation there is only nitrogen, C2H4,

and oxygen). The concentration of benzene does not develop fast enough where the resolution of the simulation can detect a smooth pattern. Once the simulation progresses (i.e. time increases) the benzene has time to form and, given the resolution of the simulation, the profile is smooth as expected. Higher resolution simulations (which means more fine meshes) would yield more smooth profiles at the beginning. Given the scope of this study however, a finer resolution is not necessary, thus the early benzene profiles should be disregarded.

One should note that the maximum C6H6 and C2H2 concentrations do not occur at a normalized horizontal coordinate of zero. This phenomenon occurs with the normalized temperature profiles. This is due to the fact that the same normalized horizontal coordinate is used to calculate the normalized temperature profiles and the C6H6 and C2H2 profiles. The normalized horizontal coordinates are based on the measured mixing layer thickness. The measured mixing layer thickness is based upon the location of the maximum temperature.

The maximum temperature locations at each HAB occur within the mixing layer. The maximum concentrations of the two species occur right at the interface of the fuel and oxidizer jets. Recall the locations of the line of maximum temperature in Fig. 42 and line representing the interface of the fuel and oxidizer

jets in Fig. 43. The location of the maximum temperatures are towards the oxidizer jet inlet while the maximum concentrations of the relevant species are towards the fuel jet inlet. Thus, when using the normalized horizontal coordinate with respect to the maximum temperature. Based on the location of the interface, it should not be surprising that the location of maximum concentrations of C6H6 and C2H2 materialize at negative values of the normalized horizontal coordinate. This means that they transpire to the left of the maximum temperature with respect to the grid as laid out in section 3.2, since the interface of the jets is to the left of the locations of the maximum temperatures.

4.3.3 Validating the Mixing Layer Thickness with Streamline

To validate the mixing layer thickness computations, the mixing layer thickness was calculated using the streamline functionality in Fluent described in detail in section 3.7. This function in Fluent allows the user to directly select streamlines in the solution and export the locations of streamlines and the time along those streamlines. The time utilized for the estimated and measured mixing layer thicknesses was derived using the solution data. It was not exported directly. The streamline functionality lets the user export the time directly on selected streamlines. Selecting the streamline at the interface of the fuel and oxidizer jets

(per section 3.7) and exporting the properties on that streamline (most notably time) provides a method of validation for the estimated and measured mixing layer calculations. In Fig. 53 the mixing layer thickness from the fuel and oxidizer jet interface streamline properties was compared to the other methods for solving the mixing layer thickness.

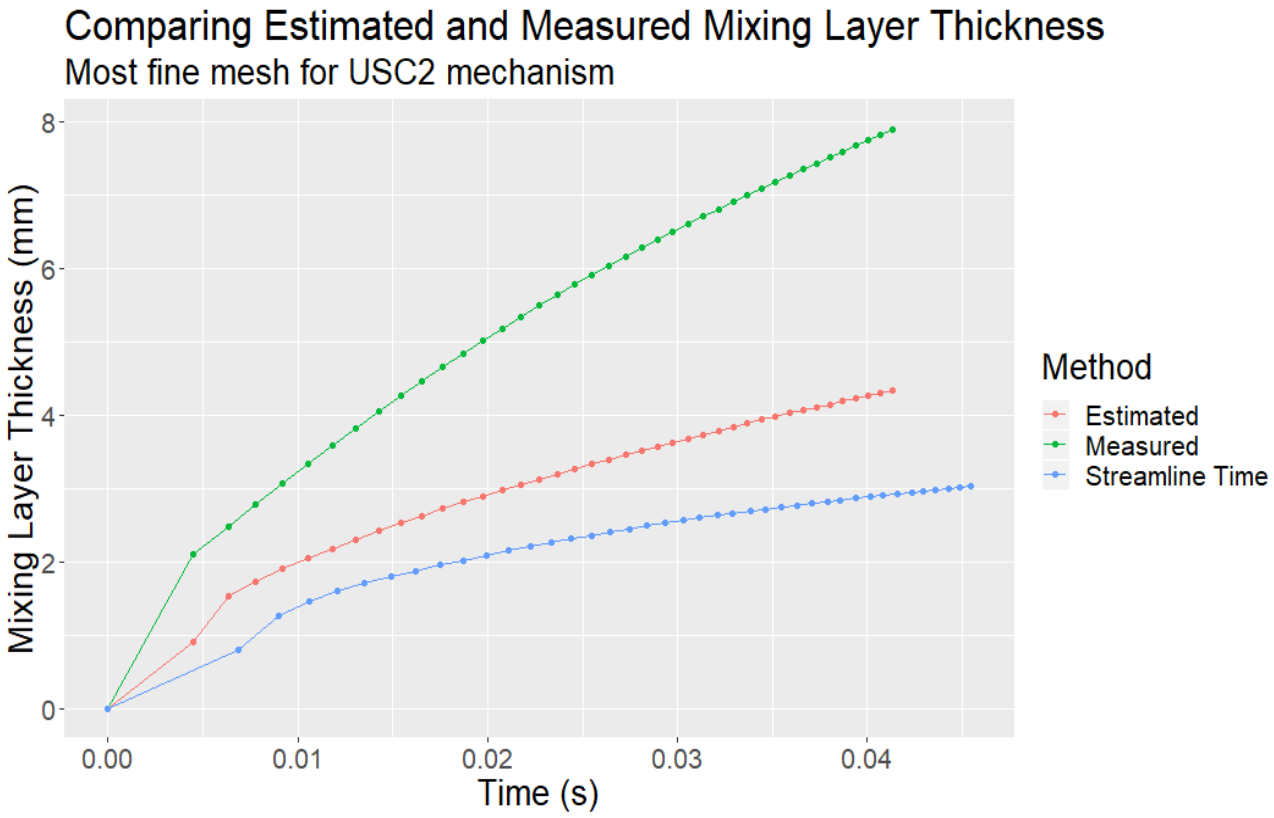

Figure 53: Comparing three mixing layer calculations for USC

As explained in section 4.2.1 and 4.3.1, estimated mixing layer thickness was expected to differ from the measured mixing layer thickness. The estimated mixing layer thickness was used to ensure that the shape of the measured mixing layer development was correct. The measured mixing layer development was more

essential because it represented the mixing layer thickness measure directly from the solution data, as opposed to the estimated mixing layer thickness which was based on a theoretical model. The mixing layer thickness shape from the streamline function should then be in agreement with the measured values because those values are based on what takes place in the simulation. To ensure this agreement, the two methods for calculating the mixing layer were compared in Fig. 54 like in Fig. 35.

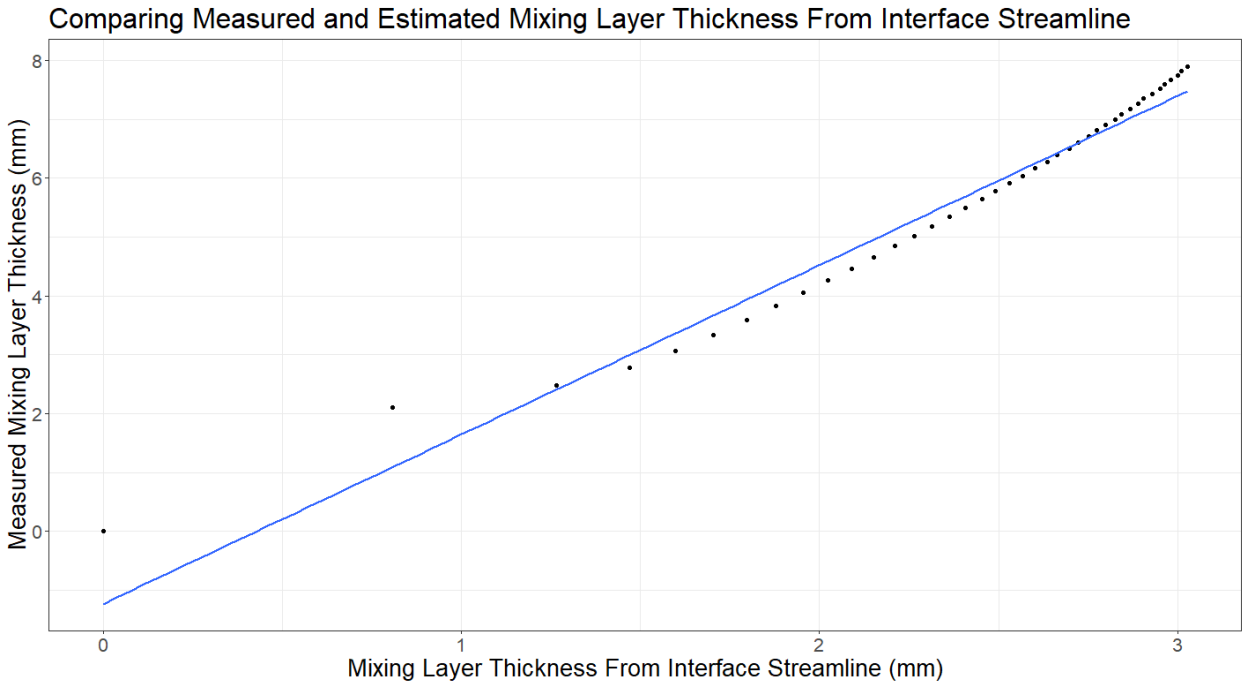

Figure 54: Comparing measured mixing layer thickness and that from the interface streamline

The magnitude of the mixing layer thicknesses from the streamline function are lower because the mixing layer thickness is based on the viscosity which is based on the temperature. The temperature on the interface streamline is about

55 to 60 percent of the temperature on the maximum temperature line. The viscosity of the mixture along the interface streamline was not directly exported from Fluent. The streamline functionality did not support it. Thus, using data from diffused nitrogen, the viscosity was derived by fitting a polynomial to the viscosity with temperature and density as the inputs. The temperature and density of the input were properties that could be exported from the streamline function. Both the estimated and measured mixing layer thickness are derived using the points on the maximum temperature line. Based on the values displayed in Fig. 54, the mixing layer development measured from the solution data and the streamline function are similar, validating the methods used to yield the measured mixing layer thickness.

4.4 Comparing the Results of the Two Mechanisms

4.4.1 Comparing the Behavior of Temperature

The results obtained modeling the flame chemistry with the USC2 and 5-Step mechanisms were compared for validating the results for the USC2 simulations and as a method for contrasting the two mechanisms (considering the differing species). The USC2 mechanism takes longer (more iterations) to converge due to the computational power required to simulate the reactions of the 100 species. The 5-Step mechanism only involves 10 species. In Fig. 55 the horizontal temperature profiles at select HABs for the USC2 and 5-Step mechanisms are overlaid.

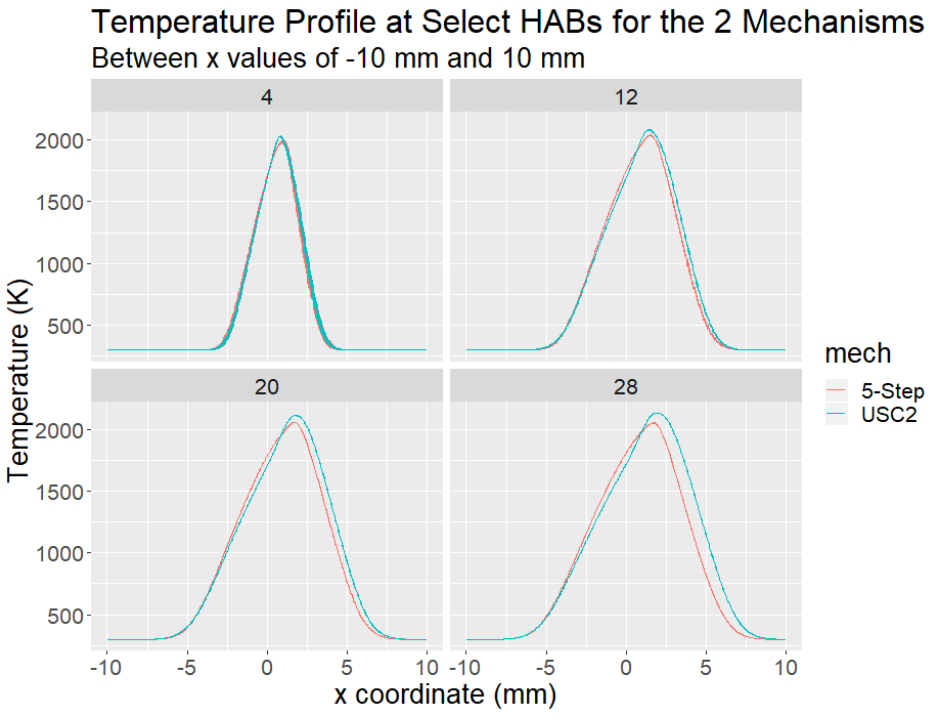

Figure 55: Comparison of temperature profiles between mechanisms

The mechanisms behave similarly (with respect to temperature) close to the inlet but diverge as the mixture develops. The maximum temperatures are similar. As the flow approaches the outlet slot at the top of the computational region, the difference in max temperatures approaches six percent relative to the USC2 mechanism temperature, with the percent difference increasing at higher HABs. The locations of the maximum temperature are also similar assessing the minimal differences due to different coupling of the chemistry with the velocity field. Fig. 56 and Fig. 57 show the differences in the maximum temperature at each HAB and the position of the maximum temperatures, respectively.

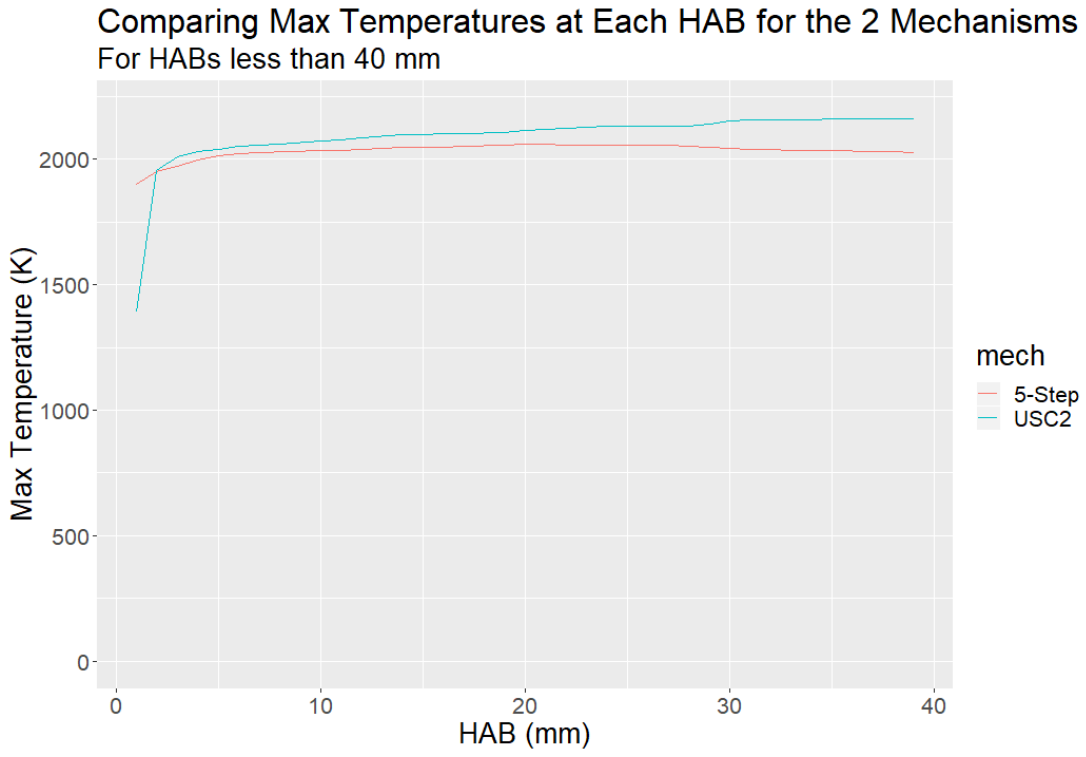

Figure 56: Comparing maximum temperatures at each HAB between 5-Step and USC

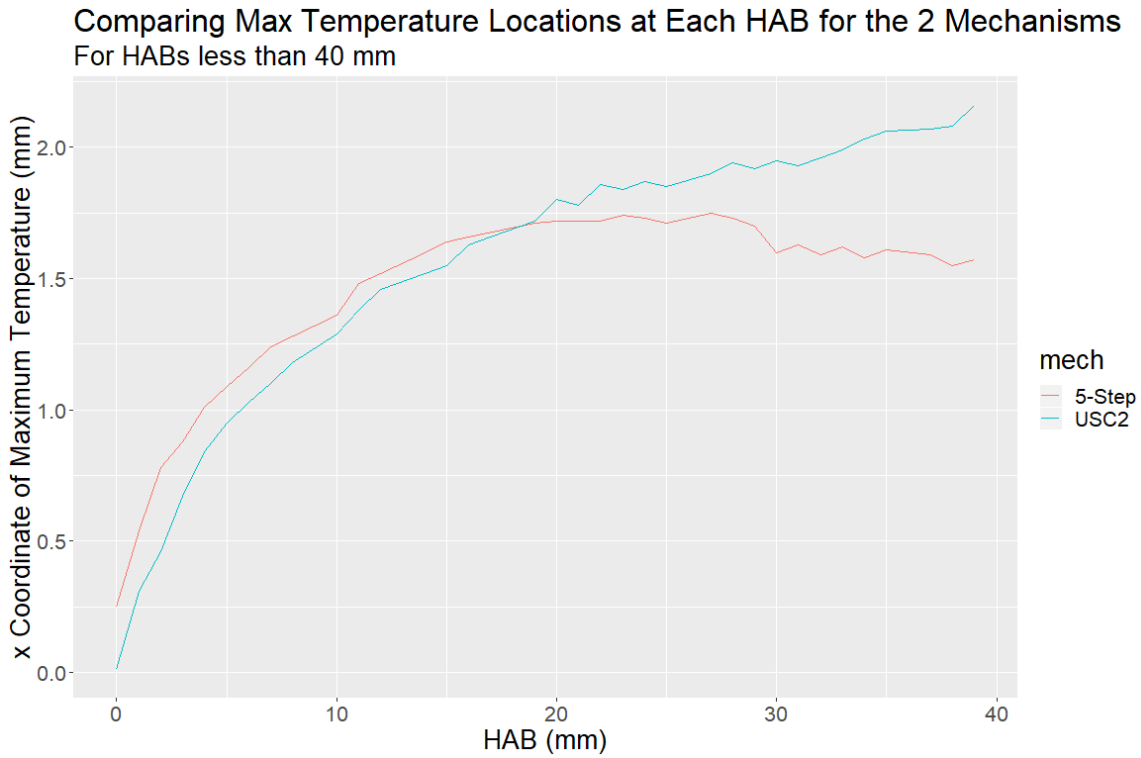

Figure 57: Comparing locations of maximum temperatures for the two mechanisms

4.4.2 Comparing Species in Each Mechanism

To further compare the solutions for the two mechanisms, the concentrations of three species in both mechanisms were observed: CO2, H2O, and CO. These species are instrumental understanding the combustion reactions. CO2 and H2O are the products of the complete oxidation reaction between the fuel (C2H4) and the oxidizer (O2), whereas CO is a product of incomplete oxidation. To investigate the differences in the mechanisms at several HABs, the horizontal profiles of concentrations of CO2 and H2O are shown in Fig. 58 and Fig. 59, respectively.

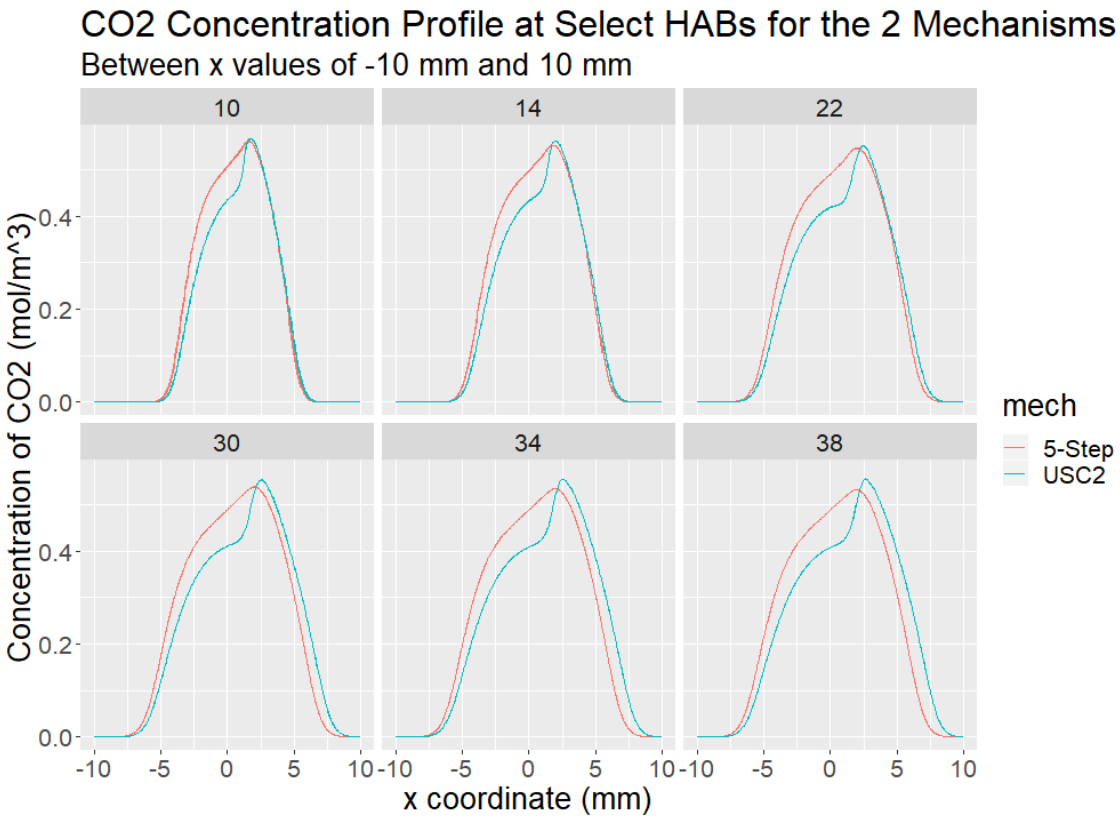

Figure 58: Comparison of CO2 profiles between the two mechanisms

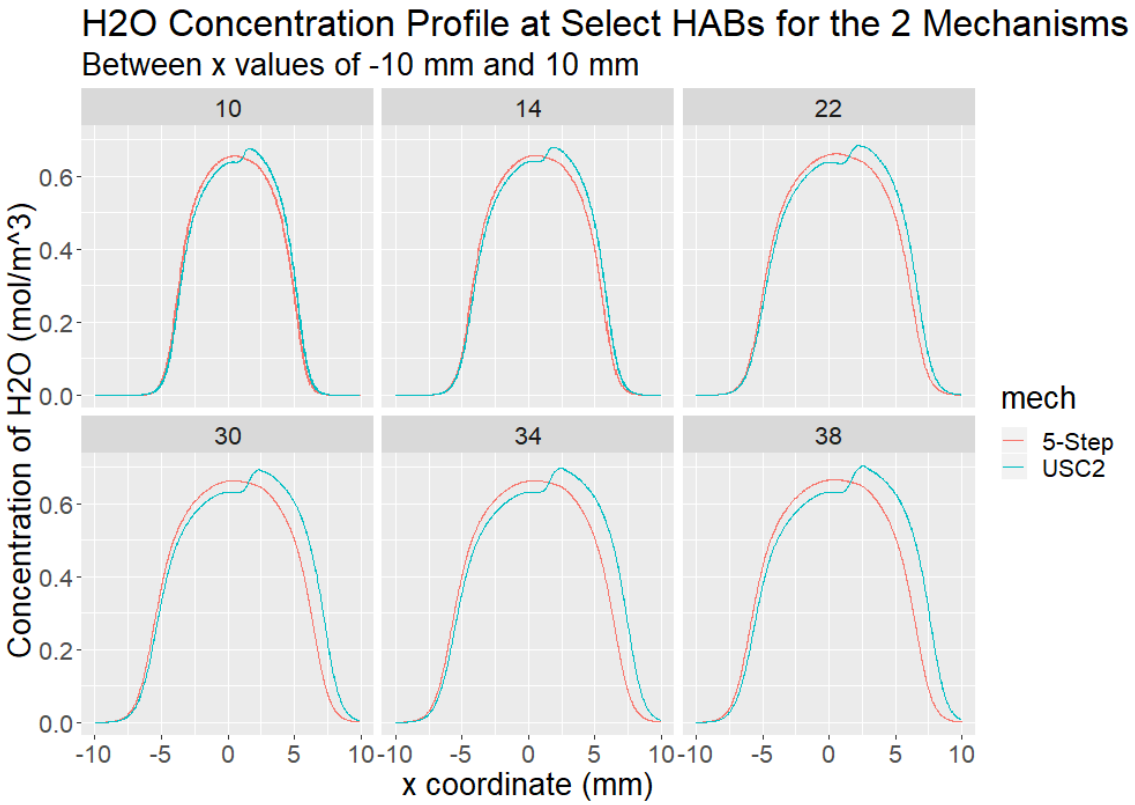

Figure 59: Comparison of H2O profiles between the two mechanisms

The formation of these two products are expectedly similar regardless of the level of detail of the used kinetic mechanism. For CO2 the maximum concentrations at each HAB are very close. The profiles show some minor differences as the flame products are convected downstream, but the magnitudes are similar since the main differences are due to changes in the locations of the maximum due to the small differences in the calculated velocity fields. The H2O profiles and maximums are also similar. The profiles diverge as the flame develops. The locations of the maximum concentrations differ slightly (by about five millimeters) but the differences in the maximum concentrations are within five percent.

Fig. 60 shows a comparison of the CO concentration profiles which highlight the improved ability to accurately model the flame chemistry with a detailed mechanism such as the USC one.

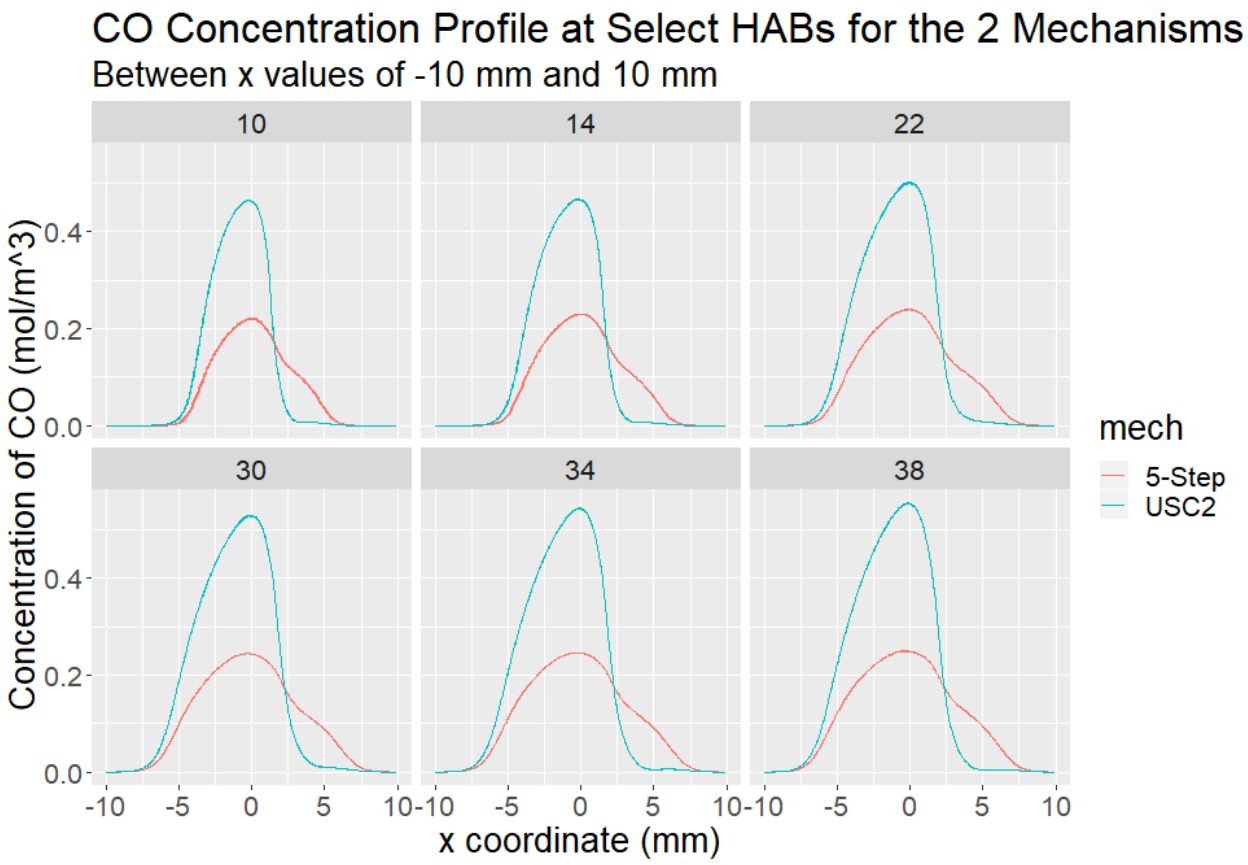

Figure 60: Comparison of CO profiles between the two mechanisms

Consistently at each of the selected HABs the concentration of CO predicted using the USC2 mechanism is approximately double that of the 5-Step mechanism because the latter overestimates the complete conversion of the fuel to complete oxidation products in the zone of the flame where CO is maximum. CO is just one byproduct of the combustion reactions.

4.5 Calculating the Rate of Soot Production

With all of the solution data processed, the rate of soot production by nucleation can be calculated with a model that fits existing experimental data in counterflow diffusion flames [Gleason, K., Carbone, F., & Gomez, A. (2018). Effect of Temperature on soot inception in highly controlled counterflow ethylene flames. Combustion and Flame, 192, 283-294.]. As alluded to above, the relevant species in calculating the rate of soot production are C2H2 and C6H6. The rate of soot production is modelled as a function of the concentration of C2H2, the concentration of C6H6, and the temperature:

$$\dot{\omega} = [C2H2][C6H6] * \exp(\frac{-E_a}{RT}) \quad (9)$$

Where, $\dot{\omega}$ is the rate of soot production in $\frac{mol}{m^3 s}$, E_a is the activation energy in $\frac{kcal}{mol}$, *R* is the universal gas constant in $\frac{kcal}{mol*K}$, and *T* is the temperature in Kelvin. $\exp(\frac{-E_a}{RT})$ is referred to as *k* and is constant for a given temperature.

This soot production rate calculation is not based directly on the simulation but is extrapolated from the solution data. Fig. 61 shows the relationship between *k* and the inverse of the temperature.

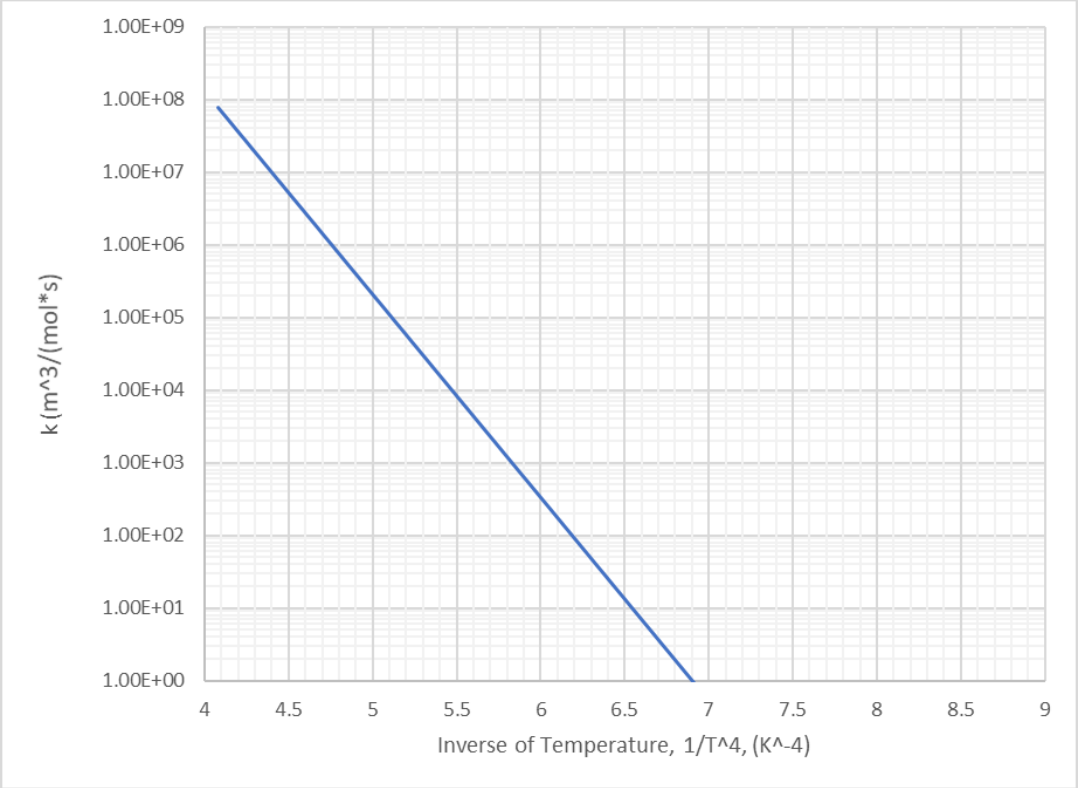

Figure 61: Relationship between concentration adjusted soot production and the inverse of the temperature

The value of *k* is largest at when the temperature is large relative to the rest of the data set. Given that the concentration of C2H2 and C6H6 increases with temperature and the rate of soot production is a function of *k* and those concentrations, the rate of soot production should also increase at higher temperatures. The maximum temperatures at each HAB occur within the mixing layer towards the side of the oxidizer jet, as detailed above. Fig. 62 shows the rates of soot production along the horizontal coordinate values of the computational area. Fig. 63 demonstrates the soot production rates at different values of temperature.

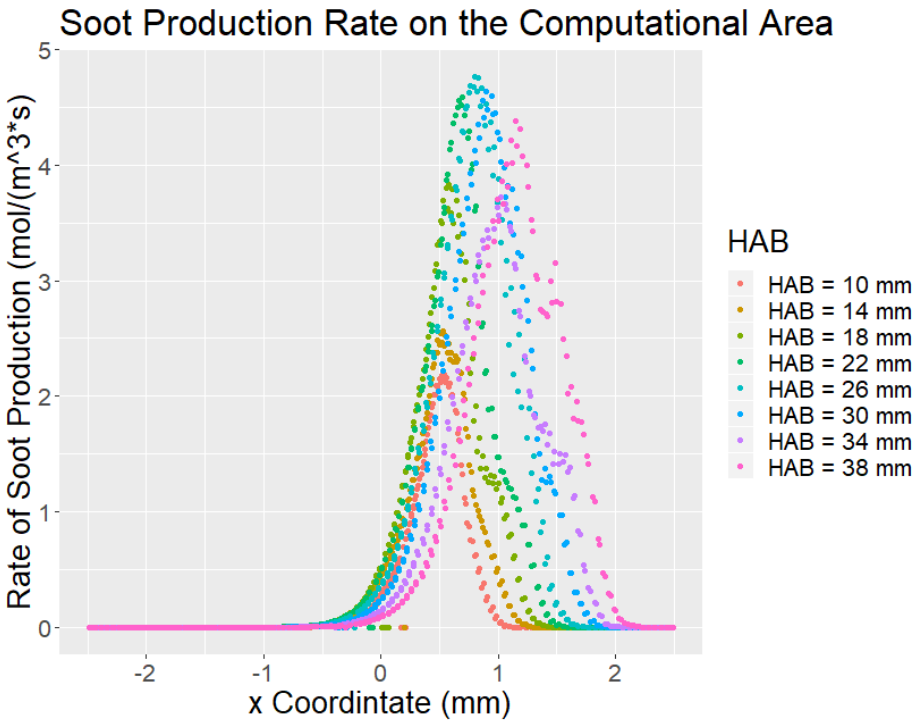

Figure 62: Soot production rates at different horizontal locations on the computational area

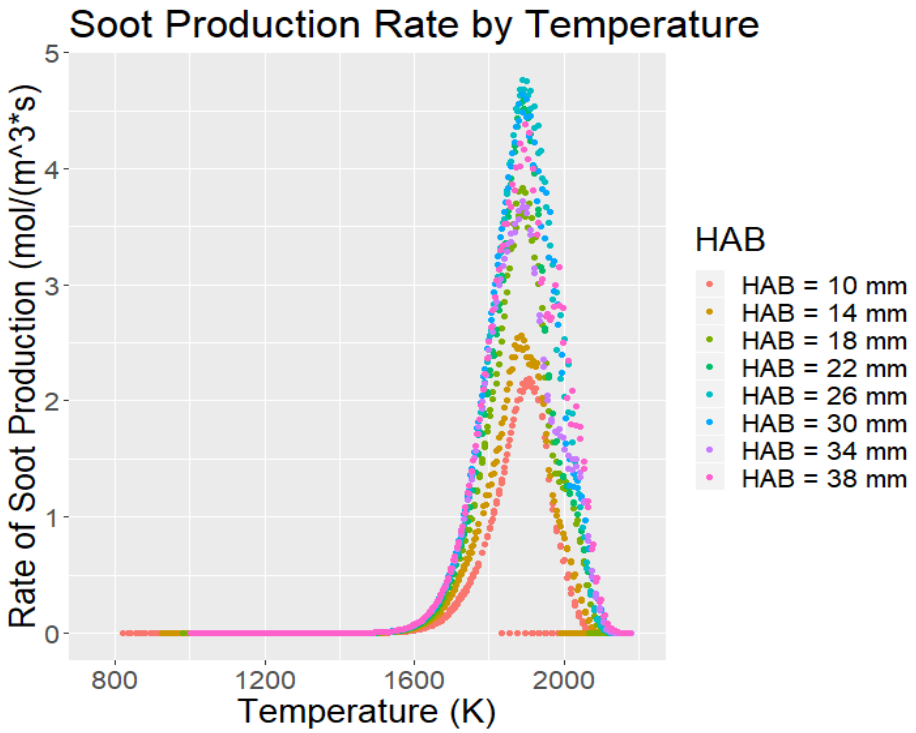

Figure 63: Soot production rate by temperature

These figures are two different representations of the same phenomenon: the rate of soot production increases with temperature because C2H2 and C6H6 is the product of high energy combustion reactions.

5. Conclusion

In this study, a new flame configuration for understanding soot inception was proposed. The dangers associated with combustion processes yielding significant soot particulate were laid out, including the health of human beings, ecosystems, and ambient atmospheric conditions.

Soot emission affects present combustion based technologies that make up for the vast majority of the energy production in the world today. The importance of leveraging combustion in energy production does not project to subside in the near future. Soot particulates, have been shown in many studies to have negative effects on human health and climate change. Exposure to soot particulates leads to health conditions such as asthma and worsen conditions such as heart and lung disease. Soot emissions also contribute to the damage of our food supply and ecosystems. The emission of soot particulate are proved to be detrimental also to air and water quality and weather patterns.

Providing a platform to study soot inception is therefore of utmost importance in improving human health and the mitigation of the effects of climate change now and in the future.

The results from this study should help researchers in the future in formulating both more robust modelling processes and physical test rigs that utilize Planar Mixing Layer (PML) flames. After processing the results of the Fluent models, the mixing layer for this flame configuration was determined to reach anywhere from five to seven millimeters. These figures are larger than those of other flame configurations, demonstrating the merits of using PML flames in future soot particulate studies. The results of the simulations also prove that PML flames yield the necessary hydrocarbon compounds to study the inception of soot particulate, mainly CO2, H2O, C2H2, and C6H6. The profiles of these species should provide an understanding of the makeup of the mixing layer for this flame configuration for future models and tests.

5.1 Recommendations for Future Work

The results of this study provide a baseline for the future study of PML flames. The 5-Step combustion mechanism (with 10 species) was used to provide the framework for processing data from the Fluent model and was used as a comparison to the USC2 combustion mechanism (with 110 species). The USC2

mechanism was considered a more realistic simulation of the combustion mechanisms present in technologies today, due to the amount of species and reactions considered and the more complex chemistry.

However, the work done in this study should not mark the end of working with PML flames and is only the first step toward the introduction of the proposed flame configuration. First, no physical tests were conducted in this study and the model introduced here will provide the guidelines to proceed with the experimental work. To ensure the feasibility of using the PML flame configuration to understand soot inception and its improvements on existing flame configurations, future physical testing of the configuration is imperative. Before widespread use of this flame configuration is acceptable, robust physical testing is required to validate the results of the Fluent models.

Furthermore, this flame configuration should be continued to be modelled by researchers in the future to identify boundary conditions for studying soot inception in a variety of flame conditions. The mesh convergence study of the USC2 mechanism showed the mesh independence of the results for pressure, temperature, velocity, and C2H2 concentration. The C6H6 concentration results were much more dependent on the size of the mesh. Due to the computational

costs and time constraint of the study, finer computation meshes for the USC2 mechanism were not used in the study. To ensure that the results for the C6H6 concentration are mesh independent, this study recommends that finer computational meshes are utilized with the PML flame configuration and USC2 combustion mechanism before any physical tests are conducted.

References

[1] J. V. J. a. S. A. Lighty, "Combustion Aerosols: Factors Governing Their Size and Composition and Implications to Human Health," *Journal of Air & Waste Management Association,* vol. 50, pp. 1565-1618, 2000.

[2] Y. a. S. H. C. Wang, "Soot formation in laminar counterflow flames," *Progress in Energy and Combustion Science,* vol. 74, pp. 142-238, 2019.

[3] EIA, "International Energy Outlook 2019," U.S. Energy Information Administration, Washington DC, 2019.

[4] United States Environmental Protection Agency, "Impacts of Diesel Emissions," Washington D.C., 2017.

[5] H. Omidvarborna, A. Kumar and D.-S. Kim, "Characterization of particulate matter emitted from transit buses fuled with B20 in idle modes," *Journal of Environmental Chemical Engineering,* pp. 2335-2342, 2014.

[6] Y. J. Kim, K. W. Kim, S. D. Kim, B. K. Lee and J. S. Han, "Fine particulate matter characteristics and its impact on the visibility impairment at two urban sites in Korea: Seoul and Inchen," *Atmospheric Environment,* vol. 40, pp. 593-605, 2006.

[7] K. A. Miller, D. S. Siscovick, L. Sheppard, K. Shephard, J. H. Sullivan, G. L. Anderson and J. D. Kaufman, "Long-Term Exposure to Air Pollution and Incidence of Cardiovascular Events in Women," *The New England Journal of Medicine,* 2007.

[8] M. Choi, A. Hamins, G. Mulholland and T. Kashiwagi, "Simultaneous optical measurement of soot volume fraction and temperature in premixed flames," *Combustion and Flame,* vol. 99, no. 1, pp. 174-186, 1994.

[9] I. Glassman, "Soot Formation in Combustion Processes," *Symposium (International) on Combustion,* vol. 22, no. 1, pp. 295-311, 1989.

[10] K. Shandilya and A. Kumar, "Particulate emissions from tailpipe during idling of public transit buses fueled with alternative fuels," *Environmental Progress & Sustainable Energy,* vol. 32, no. 4, pp. 1134-1142, 2013.

[11] D. R. Tree and K. I. Svensson, "Soot processes in compression ignition engines," *Progress in Energy and Combustion Science,* vol. 33, no. 3, pp. 272-309, 2007.

[12] H. Richter and J. Howard, "Formation of polycyclic aromatic hydrocarbons and their growth to soot- a review of chemical reaction pathways," *Process in Energy and Combustion Science,* vol. 26, no. 4-6, pp. 565-608, 2000.

[13] H. Omidvarborna, A. Kumar and D.-S. Kim, "Recent studies on soot modeling for diesel combustion," *Renewable and Sustainable Energy Reviews,* vol. 48, pp. 635-647, 2015.

[14] I. M. Kennedy, "Models of soot formation and oxidation," *Progress in Energy and Combustion Science,* vol. 23, no. 2, pp. 95-132, 1997.

[15] W. Gong and G. J. Micklow, "A multistage combustion model and soot formation model for direct-injection diesel engines," *Sage Journals,* vol. 216, no. 6, pp. 495-504, 2002.

[16] A. Kazakov and D. E. Foster, "Modeling of Soot Formation During DI Diesel Combustion Using a Multi-Step Phenomenological Model," *SAE Transcations,* vol. 107, no. 4, pp. 1016-1028, 1998.

[17] I. Khan, G. Greeves and D. Probert, "Air pollution control in transport engines," *Institution of Mechanical Engineers,* vol. 205, 1971.

[18] K. Gleason, F. Carbone and A. Gomez, "Effect of temperature on soot inception in highly controlled counterflow ethylene diffusion flames," *Combustion and Flame,* vol. 192, pp. 283-294, 2018.

[19] H. Wang and X. You, "High-Temperature Combustion Reaction Model of H2/CO/C1-C4 Compounts," Univeristy of Southern California, 2007.

[20] L. S. Taura, I. B. Ishiyaku and A. H. Kawo, "The use of a continuity equation of fluid mechanics to reduce the abnormality of the cardiovascular system: A control mechanics of the human heart," *Journal of Biophysics and Structural Biology,* vol. 4, no. 1, pp. 1-12, 2012.

[21] V. J. Katz, "The History of Stokes Theorem," *Mathematics Magazine,* vol. 52, no. 3, pp. 146-156, 1979.

Appendix

Source code for mesh convergence 5-Step

```
library(tidyverse)


# need to check the convergence of the mesh

# look at values by vertical cut or see if you can inner_join by coordinate


# first pull the data from the OG mesh, where the element size is 4e-03, number of nodes is 3933

df1 <- read_csv('3000_iterations_all_vars') %>%
  mutate(nodes = 3933,
         y_prof = ifelse(abs(`y-coordinate` - .005)/.005 <= .2, '5 mm',
                         if_else(abs(`y-coordinate` - .015)/.015 <= .1, '15 mm',
                                 if_else(abs(`y-coordinate` - .025)/.025 <= .05, '25 mm',
                                         ifelse(abs(`y-coordinate` - .035)/.035 <= .05, '35 mm',
                                                'N/A'))))) %>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)


df1$y_prof <- factor(df1$y_prof, c("5 mm", "15 mm", "25 mm", "35 mm"))


#pull the data from a new mesh, where element size is 1e-03, number of nodes is 7532

df2 <- read_csv('mesh_two_data') %>%
  mutate(nodes = 7532,
         y_prof = ifelse(abs(`y-coordinate` - .005)/.005 <= .1, '5 mm',
                         if_else(abs(`y-coordinate` - .015)/.015 <= .1, '15 mm',
                                 if_else(abs(`y-coordinate` - .025)/.025 <= .05, '25 mm',
                                         ifelse(abs(`y-coordinate` - .035)/.035 <= .05, '35 mm',
```

```
                                'N/A')))))%>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)

df2$y_prof <- factor(df2$y_prof, c("5 mm", "15 mm", "25 mm", "35 mm"))

#pull the data from a new mesh, where element size is 5e-04, number of nodes is 20493
df3 <- read_csv('mesh_three_data') %>%
  mutate(nodes = 20493,
         y_prof = ifelse(abs(`y-coordinate` - .005)/.005 <= .1, '5 mm',
                  if_else(abs(`y-coordinate` - .015)/.015 <= .1, '15 mm',
                      if_else(abs(`y-coordinate` - .025)/.025 <= .05, '25 mm',
                          ifelse(abs(`y-coordinate` - .035)/.035 <= .05, '35 mm',
                              'N/A')))))%>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)

df3$y_prof <- factor(df3$y_prof, c("5 mm", "15 mm", "25 mm", "35 mm"))

#pull the data from a new mesh, where element size is 4e-04, number of nodes is 30381
df4 <- read_csv('mesh_four_data') %>%
  mutate(nodes = 30381,
         y_prof = ifelse(abs(`y-coordinate` - .005)/.005 <= .1, '5 mm',
                  if_else(abs(`y-coordinate` - .015)/.015 <= .1, '15 mm',
                      if_else(abs(`y-coordinate` - .025)/.025 <= .05, '25 mm',
                          ifelse(abs(`y-coordinate` - .035)/.035 <= .05, '35 mm',
                              'N/A')))))%>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)
```

```
df4$y_prof <- factor(df4$y_prof, c("5 mm", "15 mm", "25 mm", "35 mm"))

#pull the data from a new mesh, where element size is 3e-04, number of nodes is 51134
df5 <- read_csv('mesh_five_data') %>%
  mutate(nodes = 51134,
         y_prof = ifelse(abs(`y-coordinate` - .005)/.005 <= .1, '5 mm',
                         if_else(abs(`y-coordinate` - .015)/.015 <= .1, '15 mm',
                                 if_else(abs(`y-coordinate` - .025)/.025 <= .05, '25 mm',
                                         ifelse(abs(`y-coordinate` - .035)/.035 <= .05, '35 mm',
                                                'N/A')))))%>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)

df5$y_prof <- factor(df5$y_prof, c("5 mm", "15 mm", "25 mm", "35 mm"))

# to test the mesh convergence, can try 1 of 2 things:
# look at the maximum values for a given property
# or can look at the values by the y_prof variable

# combine all the data
df <- bind_rows(df1, df2) %>% bind_rows(df3) %>% bind_rows(df4) %>% bind_rows(df5)
df <- df[,c(-5, -23, -24, -25)]
df <- df %>% na.omit()

# create the max temperature plot by vertical cut
df %>%
  filter(y_prof != 'N/A') %>%
  group_by(nodes, y_prof) %>%
```

```
  summarise(max_temp = max(`total-temperature`)) -> max_df

max_df$nodes <- factor(max_df$nodes, c('3933', '7532', '20493', '30381', '51134'))
max_df$y_prof <- factor(max_df$y_prof, c("5 mm", "15 mm", "25 mm", "35 mm"))

max_df %>%
  ggplot(aes(y_prof, max_temp, color = nodes, group = nodes))+
  geom_line()+
  geom_point()+
  labs(x = 'Distance from the Nozzle', y = 'Maximum Temperature (K)',
       title = 'Maximum Temperature at Each Vertical Cut by Mesh Size')

# create distribution of pressures plot (2D Mesh convergence plot)
df %>%
  ggplot(aes(pressure)) +
  stat_density(geom = 'line') +
  facet_grid(nodes~y_prof) +
  labs(x = 'Pressure (Pa)', y = 'Proportion of Nodes',
       title = 'Distribution of Pressures by Vertical Cut and Mesh Size')

# create horizontal profile of temperature values, sorted by distance from Nozzle
df$nodes <- factor(df$nodes, c('3933', '7532', '20493', '30381', '51134'))
df %>%
  filter(`total-temperature` != 0,
         x < .01, x > -.01) %>%
  ggplot(aes(x,`total-temperature`, group = nodes, color = nodes)) +
  geom_line()+
  facet_grid(nodes~y_prof) +
  labs(x = 'x coordinate (m)', y = 'Temperature (K)',
```

```
    title = 'Temperature based on Horizontal Coordinate',
    subtitle = 'Filtered by Distance from Nozzle and Mesh Size')+
  scale_color_discrete(name = "Number of Nodes")

# create horizontal profile of x velo values, sorted by distance from Nozzle
df$nodes <- factor(df$nodes, c('3933', '7532', '20493', '30381', '51134'))
df %>%
  filter(`total-temperature` != 0,
         x < .01, x > -.01) %>%
  ggplot(aes(x,`x-velocity`, group = nodes, color = nodes)) +
  geom_smooth(se=F)+
  facet_grid(nodes~y_prof) +
  labs(x = 'x coordinate (m)', y = expression(paste("x velocity   ", (  frac(m, s)  ))),
    title = 'x Velocity based on Horizontal Coordinate',
    subtitle = 'Filtered by Distance from Nozzle and Mesh Size')+
  scale_color_discrete(name = "Number of Nodes") +
  scale_x_continuous(breaks=seq(-.01,.01,.01))
# create horizontal profile of CO
df$nodes <- factor(df$nodes, c('3933', '7532', '20493', '30381', '51134'))
df %>%
  filter(`total-temperature` != 0,
         x < .01, x > -.01) %>%
  ggplot(aes(x,`molef-co`, group = nodes, color = nodes)) +
  geom_line()+
  facet_grid(nodes~y_prof) +
  labs(x = 'x coordinate (m)', y = 'Mole Fraction of CO',
    title = 'Mole Fraction of CO based on Horizontal Coordinate',
    subtitle = 'Filtered by HAB and Mesh Size')+
  scale_color_discrete(name = "Number of Nodes") +
```

```
  scale_x_continuous(breaks=seq(-.01,.01,.01))
# create horizontal profile of pressure, sorted by distance from Nozzle
df$nodes <- factor(df$nodes, c('3933', '7532', '20493', '30381', '51134'))
df %>%
  filter(`total-temperature` != 0,
         x < .01, x > -.01) %>%
  ggplot(aes(x,pressure, group = nodes, color = nodes)) +
  geom_line()+
  facet_grid(nodes~y_prof) +
  labs(x = 'x coordinate (m)', y = 'Pressure (Pa)',
       title = 'Pressure based on Horizontal Coordinate',
       subtitle = 'Filtered by Distance from Nozzle and Mesh Size')+
  scale_color_discrete(name = "Number of Nodes")



# Look at the maximum percent difference between the two finest meshes
check <- df %>%
  filter(nodes %in% c(51134, 30381)) %>%
  select(x,y,nodes, pressure) %>%
  mutate(x_round = round(x, digits = 4), y_round = round(y, digits = 4)) %>%
  select(x_round, y_round, nodes, pressure) %>%
  rename(x = x_round, y = y_round) %>%
  pivot_wider(names_from = nodes, values_from = pressure) %>%
  na.omit() %>%
  mutate(perc_diff = abs(`30381`-`51134`)/abs(`51134`))
```

Source code for initialization plots 5-Step

```
library(tidyverse)
# importing data for intialized solution
```

```
df <- read_csv('chemkin_initial') %>%
  rename(x = `x-coordinate`, # renaming columns in such a way that makes it easier to mutate and create new fields
         y = `y-coordinate`,
         o2_conc = `concentration-o2`,
         h2_conc = `concentration-h2`,
         h_conc = `concentration-h`,
         o_conc = `concentration-o`,
         oh_conc = `concentration-oh`,
         co_conc = `concentration-co`,
         h2o_conc = `concentration-h2o`,
         co2_conc = `concentration-co2`,
         n2_conc = `concentration-n2`,
         c2h4_conc = `concentration-c2h4`,
         y_velo = `y-velocity`,
         x_velo = `x-velocity`) %>%
  # mutating data by rounding the coordinates to a few decimal places
  mutate(y_round = round(y, digits = 3),
         x = round(x,digits = 5),
         y = round(y, digits = 5),
         initialized = 'Yes') %>%
  select(nodenumber, x, y, pressure, y_prof, y_round, initialized) # selecting relevent columns for convergence analysis

df$y_prof <- factor(df$y_prof, c("5 mm", "15 mm", "25 mm", "35 mm")) # ordering the discrete profile tags

# importing data for not initialized solution

# same process as before with the intialized solution

df1 <- read_csv('chemkin_no_initial') %>%
  mutate(nodes = 3933,
         y_prof = ifelse(abs(`y-coordinate` - .005)/.005 <= .1, '5 mm',
                         if_else(abs(`y-coordinate` - .015)/.015 <= .1, '15 mm',
```

```
                        if_else(abs(`y-coordinate` - .025)/.025 <= .05, '25 mm',
                                ifelse(abs(`y-coordinate` - .035)/.035 <= .05, '35 mm',
                                       'N/A'))))) %>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`,
         o2_conc = `concentration-o2`,
         h2_conc = `concentration-h2`,
         h_conc = `concentration-h`,
         o_conc = `concentration-o`,
         oh_conc = `concentration-oh`,
         co_conc = `concentration-co`,
         h2o_conc = `concentration-h2o`,
         co2_conc = `concentration-co2`,
         n2_conc = `concentration-n2`,
         c2h4_conc = `concentration-c2h4`,
         y_velo = `y-velocity`,
         x_velo = `x-velocity`) %>%
  mutate(y_round = round(y, digits = 3),
         x = round(x,digits = 4),
         y = round(y, digits = 4),
         initialized = 'No') %>%
  select(nodenumber, x, y, pressure, y_prof, y_round, initialized)

df$y_prof <- factor(df$y_prof, c("5 mm", "15 mm", "25 mm", "35 mm"))

# binding the dataframes with the initialized and uninitialized data to compare the sets

check <- bind_rows(df, df1)

# create plot demonstrating the convergence of the intialized solution with respect to the unitialized solution

# using pressure values due to their relative sensitivity
```

```
# creating separate frames for each value of y
check %>%
  filter(y_round %in% c(.006, .016, .026, .036)) %>%
  filter(pressure != 0) %>%
  ggplot(aes(x, pressure, color = initialized)) +
  geom_point() +
  facet_wrap(~factor(y_round)) +
  labs(x = 'x coordinate (m)', y = 'pressure (Pa)',
       title = 'Comparison of Initialized and Uninitialized Solutions',
       subtitle = 'Using Horizontal Pressure Profiles for Each y Coordinate') +
  theme(
    plot.title = element_text(size = 16),
    plot.subtitle = element_text(size = 14),
    axis.title = element_text(size = 14),
    legend.title = element_text(size = 16),
    legend.text = element_text(size = 16)
check <- check %>%
  select(x,y,y_round, initialized, pressure) %>%
  pivot_wider(names_from = initialized, values_from = pressure) %>%
  mutate(perc_diff = abs(Yes-No)/abs(No))
check %>%
  filter(y_round %in% c(.006, .016, .026, .036)) %>%
  ggplot(aes(x, perc_diff)) +
  geom_smooth(se = F) +
  facet_wrap(~factor(y_round)) +
  labs(x = 'x coordinate (m)', y = 'Relative Difference in Pressure',
       title = 'Relative Error Between Pressure Values for Initialized and Uninitialized Solution',
       subtitle = 'Using Horizontal Pressure Profiles for Each y Coordinate') +
  theme(
    plot.title = element_text(size = 16),
```

```
    plot.subtitle = element_text(size = 14),
    axis.title = element_text(size = 14),
    legend.title = element_text(size = 16),
    legend.text = element_text(size = 16)
  )
```

Source code for temperature profiles 5-Step

```
library(tidyverse)
library(gridExtra)
library(gghighlight)


# import data from chemkin solution intialized with equilibrium solution
df <- read_csv('chemkin_initial') %>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`,
         o2_conc = `concentration-o2`,
         h2_conc = `concentration-h2`,
         h_conc = `concentration-h`,
         o_conc = `concentration-o`,
         oh_conc = `concentration-oh`,
         co_conc = `concentration-co`,
         h2o_conc = `concentration-h2o`,
         co2_conc = `concentration-co2`,
         n2_conc = `concentration-n2`,
         c2h4_conc = `concentration-c2h4`,
         y_velo = `y-velocity`,
         x_velo = `x-velocity`) %>%
  mutate(y_round = round(y, digits = 3),
         x = round(x,digits = 5),
         y = round(y, digits = 7))
# calculate max temperature for every y value
```

```
# need to first round the numbers to get less unique numerical values


max <- df %>%
  group_by(y_round) %>%
  summarize(max_temp = max(temperature),
            min_temp = min(temperature),
            half_width_temp = min_temp + (max_temp - min_temp)/2)


df <- inner_join(df, max, by = 'y_round')


# indentify location of max temp stream and half width temp location
df <- df %>%
  mutate(max_temp_loc = ifelse(temperature == max_temp, 'Yes', 'No'),
         max_temp_loc = ifelse(y == 0, ifelse(x == 0, 'Yes', 'No'), max_temp_loc),
         half_width_temp_loc =
           ifelse(abs(temperature- half_width_temp)/temperature < .07, 'Yes', 'No'),
         half_width_temp_loc = ifelse(y == 0, ifelse(x == 0, 'Yes', 'No'), half_width_temp_loc))
# plot of temperature profiles at each y
df %>%
  filter(temperature != 0, x > -.025, x < .025) %>%
  ggplot(aes(x, temperature, color = factor(y_round))) +
  geom_line() +
  facet_wrap(~y_round) +
  theme(legend.position = 'none') +
  labs(x = 'x coordinate (m)', y = 'Temperature (K)',
       title = 'Temperature Profile at Every y Coordinate',
       subtitle = 'Between x values of -.025 m and .025 m')
```

Source code for temperature processing and delta calculations 5-Step

```
library(tidyverse)

library(gridExtra)

library(gghighlight)


# import data from chemkin solution intialized with equilibrium solution

df <- read_csv('chemkin_initial') %>%

  # rename columns so they are easier to work with

  rename(x = `x-coordinate`,

         y = `y-coordinate`,

         o2_conc = `concentration-o2`,

         h2_conc = `concentration-h2`,

         h_conc = `concentration-h`,

         o_conc = `concentration-o`,

         oh_conc = `concentration-oh`,

         co_conc = `concentration-co`,

         h2o_conc = `concentration-h2o`,

         co2_conc = `concentration-co2`,

         n2_conc = `concentration-n2`,

         c2h4_conc = `concentration-c2h4`,

         y_velo = `y-velocity`,

         x_velo = `x-velocity`) %>%

  # round off the x and y values

  mutate(y_round = round(y, digits = 3),

         x = round(x,digits = 5),

         y = round(y, digits = 7))
```

```
# calculate max temperature for every y value
# need to first round the numbers to get less unique numerical value
max <- df %>%
  group_by(y_round) %>%
  summarize(max_temp = max(temperature),
            min_temp = min(temperature),
            half_width_temp = min_temp + (max_temp-min_temp)/2)

df <- inner_join(df, max, by = 'y_round')

# indentify location of max temp stream and half width temp location
df <- df %>%
  mutate(max_temp_loc = ifelse(temperature == max_temp, 'Yes', 'No'),
         max_temp_loc = ifelse(y == 0, ifelse(x == 0, 'Yes', 'No'), max_temp_loc),
         half_width_temp_loc =
           ifelse(abs(temperature- half_width_temp)/temperature < .1, 'Yes', 'No'),
         half_width_temp_loc = ifelse(y == 0, ifelse(x == 0, 'Yes', 'No'), half_width_temp_loc))

# make plot of half width temp streams
df %>%
  ggplot(aes(x,y)) +
  geom_point() +
  gghighlight(half_width_temp_loc == 'Yes') +
  labs(title = 'Half Width Temperature Locations')

# make plot of max temp stream
ggplot(df, aes(x,y)) +
  geom_point() +
  gghighlight(max_temp_loc == 'Yes')  +
```

```
  labs(title = 'Max Temperature Locations')


# plot of temperature profiles at each y
df %>%
  filter(temperature != 0, x > -.025, x < .025) %>%
  ggplot(aes(x, temperature, color = factor(y_round))) +
  geom_line() +
  facet_wrap(~y_round) +
  theme(legend.position = 'none') +
  labs(x = 'x coordinate (m)', y = 'Temperature (K)',
       title = 'Temperature Profile at Every y Coordinate',
       subtitle = 'Between x values of -.025 m and .025 m')



# calculate the time and the mixing layer thickness (delta)
# this is what I will call computational delta in the report
time_df <- df %>%
  filter(max_temp_loc == 'Yes') %>%
  arrange(y) %>%
  mutate(delta_t = ifelse(y == 0, 0,
                          (y_round-lag(y_round))/((y_velo + lag(y_velo))/2)), # delta t is the change in position
divided by the average velo at each point
         t = cumsum(delta_t), # time is the cumulative sum of each delta 2
         delta_comp = sqrt((`viscosity-lam`/density)*t)) %>% # equation for computational mixing layer
thickness
  select(nodenumber, x, y, x_velo, y_velo, temperature,
         y_round,delta_t, t, delta_comp)


# calculating the experimental values of delta
```

```
half_df <- df %>%
  filter(half_width_temp_loc == 'Yes') %>%
  mutate(x_round = round(x, digits = 3)) %>%
  distinct(x_round, y_round, .keep_all = TRUE) %>%
  # delta is the difference in x values between the x value of the given row and the x value in either the leading or lagging row
  # check to see if the leading or lagging row is correct by making sure the y value is the same
  mutate(delta_exp = ifelse(lag(y) == y, abs(lag(x) - x), NA),
         delta_exp = ifelse(is.na(delta_exp) == TRUE, abs(lead(x) - x), delta_exp),
         delta_exp = ifelse(y == 0, 0, delta_exp)) %>%
  select(x_round, y_round, x_velo, y_velo, temperature, delta_exp) %>%
  filter(delta_exp > .001 | delta_exp == 0) %>%
  group_by(y_round) %>%
  summarise(delta_exp = mean(delta_exp))
# join experimental delta calculations with the data frame that contains computational delta and make new data frame
# which will be a new data frame joined to the main data frame
delta_df <- time_df %>%
  inner_join(half_df, by = 'y_round') %>%
  select(y_round, delta_t, t, delta_comp, delta_exp)

# need to attach the delta and time information to the main data frame
# going to need to loop through each row of data frame with delta information and append that information
# to the relevant rows in the main data frame
# I have to deal with the main data frame in chunks based on the value of the y coordinate
# first create empty list which will contain all the chunks of the main data frame
datalist = list()
# create indexing variable to specify the element in the list
n <- 1
```

```
for (i in delta_df$y_round) {
  # filter delta data frame by the given y coordinate
  df1 <- delta_df %>% filter(y_round == i)
  # join the info from the delta data frame to the relevant columns in the main data frame
  # this will create a chunk of the main data frame that is based off of the y coordinate
  df2 <- df %>% inner_join(df1, by = 'y_round')
  # store in chunk in the previously empty list and index based on the value of the indexing variable
  datalist[[n]] <- df2
  # increase the indexing variable by 1
  n <- n + 1
}
# bind all of the main data frame chunks with new info back together
tib = do.call(rbind, datalist)
# normalize the temperature by dividing it by the max temperature for the given y value
# normalize the x coordinate based on the experimental mixing layer thickness
tib <-  tib %>%
  mutate(temp_norm = temperature/max_temp,
         x_norm = ifelse(y_round == 0, x, x/delta_exp)) %>%
  filter(y < .04)

# remake the temperature profiles with the normalized x coordinate and temperature
# they should be the same
tib %>%
  filter(temperature != 0, x > -.025, x < .025) %>%
  ggplot(aes(x_norm, temp_norm, color = factor(y_round))) +
  geom_line() +
  facet_wrap(~y_round) +
  theme(legend.position = 'none') +
  labs(x = 'Normalized x Coordinate', y = 'Normalized Temperature (K)',
```

```
      title = 'Normalized Temperature Profile at Every y Coordinate',
      subtitle = 'Between x values of -.025 m and .025 m')

# create a spline for the experimental delta to interpolate what should be the correct
# shape of the data
library(splines)
# looking at the plots, 3 degrees of freedom yields the expected shape of the experimental delta data
delta_fit <- smooth.spline(tib$t, tib$delta_exp,df = 3)
tib <- tib %>%
  mutate(delta_exp_adj = predict(delta_fit, t)[[2]],
         delta_exp_adj = ifelse(y == 0, 0, delta_exp_adj))

# get the temp x locations from time_df created above
loc_df <- time_df %>%
  select(x, y_round) %>%
  rename(max_temp_x = x)

# append max temp x location data to the main data frame
tib <- tib %>%
  inner_join(loc_df, by = 'y_round')
# fix the normalized x values using new delta experimental and the x coordinate of max temp
tib <- tib %>%
  mutate(x_norm =  ifelse(y_round == 0, x, (x-max_temp_x)/delta_exp_adj))
# normalized plots again with the new normalized x
tib %>%
  filter(temperature != 0, x > -.025, x < .025) %>%
  ggplot(aes(x_norm, temp_norm, color = factor(y_round))) +
  geom_line() +
  facet_wrap(~y_round) +
```

```
  theme(legend.position = 'none') +
  labs(x = 'Normalized x Coordinate', y = 'Normalized Temperature (K)',
       title = 'Normalized Temperature Profile at Every y Coordinate',
       subtitle = 'Between x values of -.025 m and .025 m')
# plot comparing the compulational delta to the experimental delta
tib %>%
  select(t, delta_comp, delta_exp_adj) %>%
  pivot_longer(-t, names_to = 'Method', values_to = 'delta') %>%
  ggplot(aes(t, delta, color = Method)) +
  geom_point() +
  geom_line()+
  labs(x = 'Time (s)', y = 'Mixing Layer Thickness (m)',
       title = 'Comparing Computational and Experimental Mixing Layer Thickness') +
  scale_color_discrete(name = 'Method', labels = c('Computational', 'Experimental'))
```

Source code for mesh convergence USC

```
library(tidyverse)

# need to check the convergence of the mesh
# look at values by vertical cut or see if you can inner_join by coordinate

# first pull the data from the OG mesh, where the element size is 4e-03, number of nodes is 3933
df1 <- read_csv('intialized-usc2-3000-iter') %>%
  mutate(nodes = 3933,
         y_round = round(`y-coordinate`, digits = 3)) %>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)

#pull the data from a new mesh, where element size is 1e-03, number of nodes is 7532
```

```
df2 <- read_csv('initialized-usc2-3000-iter-2nd-coarse') %>%
  mutate(nodes = 7176,
         y_round = round(`y-coordinate`, digits = 3))%>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)

#pull the data from a new mesh, where element size is 5e-04, number of nodes is 20493
df3 <- read_csv('initialized-3000-iter-med-mesh') %>%
  mutate(nodes = 17326,
         y_round = round(`y-coordinate`, digits = 3))%>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)

#pull the data from a new mesh, where element size is 4e-04, number of nodes is 30381
df4 <- read_csv('initialized-3000-iter-fine-mesh') %>%
  mutate(nodes = 33756,
         y_round = round(`y-coordinate`, 3))%>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)

df5 <- read_csv('finest-mesh') %>%
  mutate(nodes = 110464,
         y_round = round(`y-coordinate`, 3))%>%
  rename(x = `x-coordinate`,
         y = `y-coordinate`)

# to test the mesh convergence, can try 1 of 2 things:
# look at the maximum values for a given property
# or can look at the values by the y_prof variable
```

```
# combine all the data
df <- bind_rows(df1, df2) %>% bind_rows(df3) %>% bind_rows(df4) %>% bind_rows(df5)



# create the max temperature plot by vertical cut
df %>%
  group_by(nodes, y_round) %>%
  summarise(max_temp = max(temperature)) %>% arrange(y_round) -> max_df

max_df$nodes <- factor(max_df$nodes, c('3933', '7176', '17326', '33756', '110464'))

max_df %>%
  ggplot(aes(factor(y_round), max_temp, color = nodes, group = nodes))+
  geom_line()+
  geom_point()+
  labs(x = 'Distance from the Nozzle (m)', y = 'Maximum Temperature (K)',
       title = 'Maximum Temperature at Each Vertical Cut by Mesh Size')+
  theme(text = element_text(size = 20)) +
  scale_x_discrete(breaks = seq(0, .048, by = 0.004))



# create distribution of pressures plot (2D Mesh convergence plot)
df %>%
  filter(y_round %in% c(.004, .016, .028, .04)) %>%
  ggplot(aes(pressure)) +
  stat_density(geom = 'line') +
  facet_grid(nodes~factor(y_round)) +
  labs(x = 'Pressure (Pa)', y = 'Proportion of Nodes',
```

```
    title = 'Distribution of Pressures by Vertical Cut and Mesh Size')+
  theme(text = element_text(size = 20))

# create horizontal profile of temperature values, sorted by distance from Nozzle
df$nodes <- factor(df$nodes, c('3933', '7176', '17326', '33756', '110464'))
df %>%
  filter(temperature != 0,
         x < .01, x > -.01) %>%
  filter(y_round %in% c(.004, .016, .028, .04)) %>%
  ggplot(aes(x,temperature, group = nodes, color = nodes)) +
  geom_line()+
  facet_grid(nodes~y_round) +
  labs(x = 'x coordinate (m)', y = 'Temperature (K)',
       title = 'Temperature based on Horizontal Coordinate',
       subtitle = 'Filtered by Distance from Nozzle and Mesh Size')+
  scale_color_discrete(name = "Number of Nodes")+
  theme(text = element_text(size = 20)) +
  scale_x_continuous(breaks = seq(-.005, .005, by = 0.005))

# create horizontal profile of x velo values, sorted by distance from Nozzle
df$nodes <- factor(df$nodes, c('3933', '7176', '17326', '33756', '110464'))
df %>%
  filter(temperature != 0) %>%
  filter(y_round %in% c(.004, .016, .028, .04)) %>%
  ggplot(aes(x,`x-velocity`, group = nodes, color = nodes)) +
  geom_line()+
  facet_grid(nodes~y_round) +
  labs(x = 'x coordinate (m)', y = 'x velocity (m/s)',
       title = 'Horizontal Velocity based on Horizontal Coordinate',
```

```
    subtitle = 'Filtered by Distance from Nozzle and Mesh Size')+
  scale_color_discrete(name = "Number of Nodes")  +
  theme(text = element_text(size = 20)) +
  scale_x_continuous(breaks = seq(-.025, .025, by = 0.025))
# create horizontal profile of C6H6
df$nodes <- factor(df$nodes, c('3933', '7176', '17326', '33756', '110464'))
df %>%
  filter(temperature != 0) %>%
  filter(y_round %in% c(.004, .016, .028, .04)) %>%
  ggplot(aes(x,`concentration-c6h6`*1000, group = nodes, color = nodes)) +
  geom_line()+
  facet_grid(nodes~y_round) +
  labs(x = 'x coordinate (m)', y = 'Concentration of C6H6 (mol/m^3)',
    title = 'Concentration of C6H6 based on Horizontal Coordinate',
    subtitle = 'Filtered by HAB and Mesh Size')+
  scale_color_discrete(name = "Number of Nodes") +
  theme(text = element_text(size = 20)) +
  scale_x_continuous(breaks = seq(-.025, .025, by = 0.025))
# create horizontal profile of pressure, sorted by distance from Nozzle
df$nodes <- factor(df$nodes, c('3933', '7176', '17326', '33756', '110464'))
df %>%
  filter(temperature != 0) %>%
  filter(y_round %in% c(.004, .016, .028, .04)) %>%
  ggplot(aes(x,pressure, group = nodes, color = nodes)) +
  geom_line()+
  facet_grid(nodes~y_round) +
  labs(x = 'x coordinate (m)', y = 'Pressure (Pa)',
    title = 'Pressure based on Horizontal Coordinate',
    subtitle = 'Filtered by Distance from Nozzle and Mesh Size')+
```

```
  scale_color_discrete(name = "Number of Nodes")+
  theme(text = element_text(size = 20)) +
  scale_x_continuous(breaks = seq(-.025, .025, by = 0.025))

# made max convergence plot for c6h6
df %>%
  group_by(nodes, y_round) %>%
  summarise(max_ben = max(`concentration-c6h6`)) %>% arrange(y_round) -> max_df

max_df$nodes <- factor(max_df$nodes, c('3933', '7176', '17326', '33756', '110464'))
max_df %>%
  ggplot(aes(factor(y_round), max_ben*1000, color = nodes, group = nodes))+
  geom_line()+
  geom_point()+
  labs(x = 'Distance from the Nozzle (m)', y = 'Maximum C6H6 Concentration (mol/m^3)',
       title = 'Maximum C6H6 Concentration at Each Vertical Cut by Mesh Size')+
  theme(text = element_text(size = 20)) +
  scale_x_discrete(breaks = seq(0, .048, by = 0.004))
# made max convergence plot for c2h2
df %>%
  group_by(nodes, y_round) %>%
  summarise(max_ben = max(`concentration-c2h2`)) %>% arrange(y_round) -> max_df

max_df$nodes <- factor(max_df$nodes, c('3933', '7176', '17326', '33756', '110464'))
max_df %>%
  ggplot(aes(factor(y_round), max_ben*1000, color = nodes, group = nodes))+
  geom_line()+
  geom_point()+
  labs(x = 'Distance from the Nozzle (m)', y = 'Maximum C2H2 Concentration (mol/m^3)',
```

```
    title = 'Maximum C2H2 Concentration at Each Vertical Cut by Mesh Size')+
  theme(text = element_text(size = 20)) +
  scale_x_discrete(breaks = seq(0, .048, by = 0.004))
```

Source code for processing finest mesh USC

```
library(tidyverse)
library(gridExtra)
library(gghighlight)

# import data from chemkin solution intialized with equilibrium solution
df <- read_csv('finest-mesh') %>%
  # rename columns so they are easier to work with
  rename(x = `x-coordinate`,
         y = `y-coordinate`,
         o2_conc = `concentration-o2`,
         c2h4_conc = `concentration-c2h4`,
         c2h2_conc = `concentration-c2h2`,
         c6h6_conc = `concentration-c6h6`,
         y_velo = `y-velocity`,
         x_velo = `x-velocity`) %>%
  # round off the x and y values
  mutate(y_round = round(y, digits = 3),
         x = round(x,digits = 5),
         y = round(y, digits = 7))

# calculate max temperature for every y value
# need to first round the numbers to get less unique numerical value
max <- df %>%
  filter(temperature != 0) %>%
```

```
  group_by(y_round) %>%
  summarize(max_temp = max(temperature),
            min_temp = min(temperature),
            half_width_temp = min_temp + (max_temp-min_temp)/2)

df <- inner_join(df, max, by = 'y_round')

# indentify location of max temp stream and half width temp location
df <- df %>%
  mutate(max_temp_loc = ifelse(temperature == max_temp, 'Yes', 'No'),
         max_temp_loc = ifelse(y == 0, ifelse(x == .00001, 'Yes', 'No'), max_temp_loc),
         half_width_temp_loc =
           ifelse(abs(temperature- half_width_temp)/temperature <= 0.05, 'Yes', 'No'),
         half_width_temp_loc = ifelse(y == 0, ifelse(x == .00001, 'Yes', 'No'), half_width_temp_loc))

# make plot of half width temp streams
df %>%
  ggplot(aes(x*1000,y*1000)) +
  geom_point() +
  gghighlight(half_width_temp_loc == 'Yes') +
  labs(title = 'Half Width Temperature Locations',
       x = 'x (mm)', y = 'HAB (mm)')  +
  theme(text = element_text(size = 20))


# make plot of max temp stream
ggplot(df, aes(x*1000,y*1000)) +
  geom_point() +
  gghighlight(max_temp_loc == 'Yes')  +
```

```
  labs(title = 'Max Temperature Locations',
       x = 'x (mm)', y = 'HAB (mm)')+
  theme(text = element_text(size = 20))

# plot of temperature profiles at each y
df %>%
  mutate(y_round = y_round*1000) %>%
  filter(temperature != 0, x > -.01, x < .01) %>%
  filter(y_round %in% seq(from = 10, to = 40, by = 4)) %>%
  ggplot(aes(x*1000, temperature, color = factor(y_round))) +
  geom_line() +
  labs(x = 'x coordinate (mm)', y = 'Temperature (K)',
       title = 'Temperature Profile at Select HABs',
       subtitle = 'Between x values of -10 mm and 10 mm',
       color = 'HAB (mm)')+
  theme(text = element_text(size = 20))


# calculate the time and the mixing layer thickness (delta)
# this is what I will call computational delta in the report
time_df <- df %>%
  filter(max_temp_loc == 'Yes') %>%
  arrange(y) %>%
  mutate(delta_t = ifelse(y_round == 0, 0,
                          (y_round-lag(y_round))/((y_velo + lag(y_velo))/2)), # delta t is the change in position divided by the average velo at each point
         t = cumsum(delta_t), # time is the cumulative sum of each delta 2
         delta_comp = sqrt((`viscosity-lam`/density)*t)) %>% # equation for computational mixing layer thickness
```

```
  select(nodenumber, x, y, x_velo, y_velo, temperature,
         y_round,delta_t, t, delta_comp)


# calculating the experimental values of delta
half_df <- df %>%
  filter(half_width_temp_loc == 'Yes') %>%
  mutate(x_round = round(x, digits = 3)) %>%
  distinct(x_round, y_round, .keep_all = TRUE) %>%
  arrange(y) %>%
  # delta is the difference in x values between the x value of the given row and the x value in either the leading or lagging row
  # check to see if the leading or lagging row is correct by making sure the y value is the same
  mutate(delta_exp = ifelse(lag(y) == y, abs(lag(x) - x), NA),
         delta_exp = ifelse(is.na(delta_exp) == TRUE, abs(lead(x) - x), delta_exp),
         delta_exp = ifelse(y_round == 0, 0, delta_exp)) %>%
  select(x_round, y_round, x_velo, y_velo, temperature, delta_exp) %>%
  filter(delta_exp > .0001 | delta_exp == 0) %>%
  group_by(y_round) %>%
  summarise(delta_exp = mean(delta_exp))
# join experimental delta calculations with the data frame that contains computational delta and make new data frame
# which will be a new data frame joined to the main data frame
delta_df <- time_df %>%
  inner_join(half_df, by = 'y_round') %>%
  select(y_round, delta_t, t, delta_comp, delta_exp) %>%
  filter(lead(delta_exp) != 0) %>%
  mutate(delta_exp = ifelse(t == 0, delta_exp, ifelse(t < .02, delta_comp*1.7, delta_comp*1.8)))
```

```
# need to attach the delta and time information to the main data frame

# going to need to loop through each row of data frame with delta information and append that information

# to the relevant rows in the main data frame

# I have to deal with the main data frame in chunks based on the value of the y coordinate

# first create empty list which will contain all the chunks of the main data frame

datalist = list()

# create indexing variable to specify the element in the list

n <- 1

for (i in delta_df$y_round) {

  # filter delta data frame by the given y coordinate

  df1 <- delta_df %>% filter(y_round == i)

  # join the info from the delta data frame to the relevant columns in the main data frame

  # this will create a chunk of the main data frame that is based off of the y coordinate

  df2 <- df %>% inner_join(df1, by = 'y_round')

  # store in chunk in the previously empty list and index based on the value of the indexing variable

  datalist[[n]] <- df2

  # increase the indexing variable by 1

  n <- n + 1

}

# bind all of the main data frame chunks with new info back together

tib = do.call(rbind, datalist)

# normalize the temperature by dividing it by the max temperature for the given y value

# normalize the x coordinate based on the experimental mixing layer thickness

tib <-  tib %>%

  mutate(temp_norm = temperature/max_temp,

         x_norm = ifelse(y_round == 0, x, x/delta_exp)) %>%

  filter(y < .045)
```

```
# remake the temperature profiles with the normalized x coordinate and temperature
# they should be the same
tib %>%
  filter(temperature != 0, x_norm > -2.5, x_norm < 2.5) %>%
  filter(y_round %in% c(0, .004, .008, .012, .016, .02, .024, .028, .032, .036)) %>%
  ggplot(aes(x_norm, temp_norm, color = factor(y_round))) +
  geom_line() +
  facet_wrap(~y_round) +
  theme(legend.position = 'none') +
  labs(x = 'Normalized x Coordinate', y = 'Normalized Temperature (K)',
       title = 'Normalized Temperature Profile at Every y Coordinate',
       subtitle = 'Between normalized x values of -2.5 and 2.5') +
  xlim(c(-2,2))+
  theme(text = element_text(size = 20))

# create a spline for the experimental delta to interpolate what should be the correct
# shape of the data
library(splines)
# looking at the plots, 3 degrees of freedom yields the expected shape of the experimental delta data
delta_fit <- smooth.spline(tib$t, tib$delta_exp,df = 3)
tib <- tib %>%
  mutate(delta_exp_adj = predict(delta_fit, t)[[2]],
         delta_exp_adj = ifelse(y_round == 0, 0, delta_exp_adj))

# get the temp x locations from time_df created above
loc_df <- time_df %>%
  select(x, y_round) %>%
  rename(max_temp_x = x)
```

```
# append max temp x location data to the main data frame
tib <- tib %>%
  inner_join(loc_df, by = 'y_round')
# fix the normalized x values using new delta experimental and the x coordinate of max temp
tib <- tib %>%
  mutate(x_norm =  ifelse(y_round == 0, x, (x-max_temp_x)/delta_exp_adj))

# get the locations of the max benzene concentration
max_b <- tib %>%
  group_by(y_round) %>%
  summarise(max_ben = max(c6h6_conc),
            min_ben = min(c6h6_conc),
            half_width_ben = min_ben + (max_ben-min_ben)/2)
# append this max benzene location data to main data frame and normalize benzene concentration
tib <- tib %>%
  inner_join(max_b, by = 'y_round') %>%
  mutate(ben_norm = c6h6_conc/max_ben)
# get the locations of the max c2h2 concentration
max_a <- tib %>%
  group_by(y_round) %>%
  summarise(max_ac = max(c2h2_conc),
            min_ac = min(c2h2_conc),
            half_width_ac = min_ac + (max_ac-min_ac)/2)
# append this max c2h2 location data to main data frame and normalize c2h2 concentration
tib <- tib %>%
  inner_join(max_a, by = 'y_round') %>%
  mutate(ac_norm = c2h2_conc/max_ac)
```

```
# normalized plots again with the new normalized x
tib %>%
  filter(temperature != 0, x_norm <= 1.5, x_norm >= -1.5) %>%
  filter(y_round %in% seq(from = .003, to = .04, by = .004)) %>%
  ggplot(aes(x_norm, temp_norm, color = factor(y_round))) +
  geom_line() +
  facet_wrap(y_round~.)+
  theme(legend.position = 'none') +
  labs(x = 'Normalized x Coordinate', y = 'Normalized Temperature',
       title = 'Normalized Temperature Profile at Select HAB',
       subtitle = 'Between normalized x values of -1.5 and 1.5',
       color = 'HAB (m)')+
  theme(text = element_text(size = 20))

# normalized benzene profiles
tib %>%
  mutate(y_round = y_round*1000) %>%
  filter(temperature != 0, x_norm <= 1.5, x_norm >= -1.5) %>%
  filter(y_round %in% seq(from = 11, to = 40, by = 4)) %>%
  ggplot(aes(x_norm, ben_norm, color = factor(y_round))) +
  geom_line() +
  facet_wrap(y_round~.)+
  theme(legend.position = 'none') +
  labs(x = 'Normalized x Coordinate', y = 'Normalized C6H6 Profile',
       title = 'Normalized C6H6 Concentration Profile at Select HAB',
       subtitle = 'Between normalized x values of -1.5 and 1.5',
       color = 'HAB (mm)')+
  theme(text = element_text(size = 20))
# normalized c2h2 profiles
```

```
tib %>%
  mutate(y_round = y_round*1000) %>%
  filter(temperature != 0, x_norm <= 1.5, x_norm >= -1.5) %>%
  filter(y_round %in% seq(from = 11, to = 40, by = 4)) %>%
  ggplot(aes(x_norm, ac_norm, color = factor(y_round))) +
  geom_line() +
  facet_wrap(y_round~.)+
  theme(legend.position = 'none') +
  labs(x = 'Normalized x Coordinate', y = 'Normalized C2H2 Profile',
       title = 'Normalized C2H2 Concentration Profile at Select HAB',
       subtitle = 'Between normalized x values of -1.5 and 1.5',
       color = 'HAB (mm)')+
  theme(text = element_text(size = 20))
# plot comparing the compulational delta to the experimental delta
tib %>%
  filter(y <= .04) %>%
  select(t, delta_comp, delta_exp_adj) %>%
  pivot_longer(-t, names_to = 'Method', values_to = 'delta') %>%
  ggplot(aes(t, delta*1000, color = Method)) +
  geom_point() +
  geom_line()+
  labs(x = 'Time (s)', y = 'Mixing Layer Thickness (mm)',
       title = 'Comparing Estimated and Measured Mixing Layer Thickness',
       subtitle = 'Most fine mesh for USC2 mechanism') +
  scale_color_discrete(name = 'Method', labels = c('Estimated', 'Measured'))+
  theme(text = element_text(size = 20))

# compare computational delta to experimental against one another
tib %>%
```

```
  filter(y <= .04) %>%
  ggplot(aes(delta_comp*1000, delta_exp_adj*1000)) +
  geom_point() +
  geom_smooth(method = 'lm', se = F) +
  theme_bw()+
  labs(x = 'Estimated Mixing Layer Thickness (mm)',
       y = 'Measured Mixing Layer Thickness (mm)',
       title = 'Comparing Measured and Estimated Mixing Layer Thickness')+
  theme(text = element_text(size = 20))

# import streamline data of interface from Fluent
stream <- read_csv('streamline-data.csv') %>%
  mutate(y_round = round(y, digits = 3)) %>%
  rename(stream_time = time,
         temperature = `Temperature [ K ]`) %>%
  select(y_round, stream_time, temperature, density) %>%
  group_by(y_round) %>%
  summarise(stream_time = mean(stream_time),
            temperature = mean(temperature),
            density = mean(density))

# model the nitrogen viscoisty to get the streamline properties
# use the density and temperature
nitro <- readxl::read_xlsx('NitrogenProperties.xlsx')
nitro.fit <- lm(viscosity ~ poly(temperature, 3) + poly(density, 3), data = nitro)
nitro$x_viscosity <- predict(nitro.fit, newdata = nitro)
nitro <- nitro %>%
  select(temperature, density, viscosity, x_viscosity) %>%
  mutate(rel_error = abs(viscosity-x_viscosity)/viscosity)
```

```
# apply model to stream data
stream <- stream %>%
  mutate(stream_time = ifelse(y_round == 0, 0, stream_time)) %>%
  mutate(viscosity = predict(nitro.fit, newdata = stream)) %>%
  rename(stream_temp = temperature,
         stream_denisty = density,
         stream_viscosity = viscosity) %>%
  mutate(stream_ml = sqrt(stream_viscosity*stream_time))

# append stream time to main data frame
tib <- tib %>%
  inner_join(stream, by = 'y_round') %>%
  mutate(stream_time = ifelse(y_round == 0, 0, stream_time))

# remake mixing layer development plot with stream time
# plot comparing the compulational delta to the experimental delta
tib %>%
  filter(y <= .04) %>%
  select(t, delta_comp, delta_exp_adj) %>%
  pivot_longer(-t, names_to = 'Method', values_to = 'delta') %>%
  ggplot(aes(t, delta*1000, color = Method)) +
  geom_point() +
  geom_line()+
  labs(x = 'Time (s)', y = 'Mixing Layer Thickness (mm)',
       title = 'Comparing Estimated and Measured Mixing Layer Thickness',
       subtitle = 'Most fine mesh for USC2 mechanism') +
  scale_color_discrete(name = 'Method', labels = c('Estimated', 'Measured'))+
  theme(text = element_text(size = 20)) -> p
```

```
dat <- tib %>%
  filter(y <= .04) %>%
  select(stream_time, delta_comp, delta_exp_adj, stream_ml) %>%
  pivot_longer(-stream_time, names_to = 'Method', values_to = 'delta') %>%
  filter(Method == 'stream_ml') %>%
  mutate(Method = 'Streamline Time')



p +
  geom_point(data = dat ,aes(stream_time, delta*1000)) +
  geom_line(data = dat, aes(stream_time, delta*1000)) +
  scale_color_discrete(name = 'Method', labels = c('Estimated', 'Measured', 'Streamline Time'))

# make c2h2 and c6h6 profs
df %>%
  filter(temperature != 0, x > -.01, x < .01) %>%
  filter(y_round %in% seq(from = .003, to = .04, by = .004)) %>%
  ggplot(aes(x*1000, c6h6_conc*1000, color = factor(y_round))) +
  geom_line() +
  facet_wrap(~y_round) +
  theme(legend.position = 'none') +
  labs(x = 'x coordinate (mm)', y = 'C6H6 Concentration (mol/m^3)',
       title = 'C6H6 Concentration Profile at Select HABs',
       subtitle = 'Between x values of -10 mm and 10 mm')+
  theme(text = element_text(size = 20))
df %>%
  filter(temperature != 0, x > -.01, x < .01) %>%
  filter(y_round %in% seq(from = .003, to = .04, by = .004)) %>%
  ggplot(aes(x*1000, c2h2_conc*1000, color = factor(y_round))) +
```

```
  geom_line() +
  facet_wrap(~y_round) +
  theme(legend.position = 'none') +
  labs(x = 'x coordinate (mm)', y = 'C2H2 Concentration (mol/m^3)',
       title = 'C2H2 Concentration Profile at Select HABs',
       subtitle = 'Between x values of -10 mm and 10 mm')+
  theme(text = element_text(size = 20))

# compare streamline time mixing layer to measured mxing layer
dat <- tib %>%
  filter(y <= .04) %>%
  select(stream_time, delta_comp, delta_exp_adj, stream_ml)

dat %>%
  ggplot(aes(stream_ml*1000, delta_exp_adj*1000))+
  geom_point() +
  geom_smooth(method = 'lm', se = F)+
  theme_bw()+
  labs(x = 'Mixing Layer Thickness From Interface Streamline (mm)',
       y = 'Measured Mixing Layer Thickness (mm)',
       title = 'Comparing Measured and Estimated Mixing Layer Thickness From Interface Streamline')+
  theme(text = element_text(size = 20))
```

Source code for comparing mechanism results

```
library(tidyverse)


# identify the mech in each data frame

tib <- tib %>%

  mutate(mech = 'usc')

tib_1 <- tib_1 %>%

  mutate(mech = 'five_step')


# select the variables to use for the comparison in each data frame

tib <- tib %>%

  select(mech, x, y, y_round, temperature, x_velo, t, delta_comp, delta_exp_adj, max_temp,
max_temp_loc, half_width_temp, half_width_temp_loc,

         x_norm, temp_norm, `concentration-co`, `concentration-co2`, `concentration-h2o`) %>%

  rename(co_conc = `concentration-co`,

         co2_conc = `concentration-co2`,

         h2o_conc = `concentration-h2o`)


tib_1 <- tib_1 %>%

  select(mech, x, y, y_round, temperature, x_velo, t, delta_comp, delta_exp_adj, max_temp,
max_temp_loc, half_width_temp, half_width_temp_loc,

         x_norm, temp_norm, co_conc, co2_conc, h2o_conc)


# create a data frame with both of the mechs

df <- tib %>%

  bind_rows(tib_1) %>%

  filter(y_round %in% seq(from = 0, to = .04, by = .001)) %>%

  mutate(y_round = y_round*1000)
```

```
# get the x locations of the max coordinates
max <- df %>%
  filter(max_temp_loc == 'Yes') %>%
  filter(x != 0) %>%
  distinct(x, y_round, .keep_all = TRUE) %>%
  select(x, y_round, mech) %>%
  rename(max_temp_coord = x)

# make the temp profiles comparing the two mechs
df %>%
  filter(temperature != 0, x > -.01, x < .01) %>%
  filter(y_round %in% seq(from = 4, to = 40, by = 8)) %>%
  ggplot(aes(x*1000, temperature, color = mech)) +
  geom_line() +
  facet_wrap(~y_round) +
  labs(x = 'x coordinate (mm)', y = 'Temperature (K)',
       title = 'Temperature Profile at Select HABs for the 2 Mechanisms',
       subtitle = 'Between x values of -10 mm and 10 mm')+
  theme(text = element_text(size = 20)) +
  scale_color_discrete(labels = c('5-Step', 'USC2'))

# compare locations of the max temps
df %>%
  filter(y_round < 40) %>%
  ggplot(aes(y_round, max_temp, color = mech)) +
  geom_line() +
  labs(x = 'HAB (mm)', y = 'Max Temperature (K)',
       title = 'Comparing Max Temperatures at Each HAB for the 2 Mechanisms',
       subtitle = 'For HABs less than 40 mm')+
```

```
  theme(text = element_text(size = 20)) +
  scale_color_discrete(labels = c('5-Step', 'USC2')) +
  ylim(c(1950, 2200))

# similar plot but with x coordinate of max temp
max %>%
  filter(y_round < .04) %>%
  ggplot(aes(y_round*1000, max_temp_coord*1000, color = mech)) +
  geom_line() +
  labs(x = 'HAB (mm)', y = 'x Coordinate of Maximum Temperature (mm)',
       title = 'Comparing Max Temperature Locations at Each HAB for the 2 Mechanisms',
       subtitle = 'For HABs less than 40 mm')+
  theme(text = element_text(size = 20)) +
  scale_color_discrete(labels = c('5-Step', 'USC2'))


# make the x_velo profiles comparing the two mechs
df %>%
  filter(temperature != 0, x > -.01, x < .01) %>%
  filter(y_round %in% seq(from = 4, to = 40, by = 8)) %>%
  ggplot(aes(x*1000, x_velo, color = mech)) +
  geom_line() +
  facet_wrap(~y_round) +
  labs(x = 'x coordinate (mm)', y = 'Horizontal Velocity (m/s)',
       title = 'Horizontal Velocity Profile at Select HABs for the 2 Mechanisms',
       subtitle = 'Between x values of -10 mm and 10 mm')+
  theme(text = element_text(size = 20)) +
  scale_color_discrete(labels = c('5-Step', 'USC2'))
```

```
# comparing concentrations of key species in both mechanisms
df %>%
  filter(temperature != 0, x > -.01, x < .01) %>%
  filter(y_round %in% seq(from = 10, to = 40, by = 4)) %>%
  ggplot(aes(x*1000, co_conc*1000, color = mech)) +
  geom_line() +
  facet_wrap(~y_round) +
  labs(x = 'x coordinate (mm)', y = 'Concentration of CO (mol/m^3)',
       title = 'CO Concentration Profile at Select HABs for the 2 Mechanisms',
       subtitle = 'Between x values of -10 mm and 10 mm')+
  theme(text = element_text(size = 20)) +
  scale_color_discrete(labels = c('5-Step', 'USC2'))

df %>%
  filter(temperature != 0, x > -.01, x < .01) %>%
  filter(y_round %in% seq(from = 10, to = 40, by = 4)) %>%
  ggplot(aes(x*1000, co2_conc*1000, color = mech)) +
  geom_line() +
  facet_wrap(~y_round) +
  labs(x = 'x coordinate (mm)', y = 'Concentration of CO2 (mol/m^3)',
       title = 'CO2 Concentration Profile at Select HABs for the 2 Mechanisms',
       subtitle = 'Between x values of -10 mm and 10 mm')+
  theme(text = element_text(size = 20)) +
  scale_color_discrete(labels = c('5-Step', 'USC2'))

df %>%
  filter(temperature != 0, x > -.01, x < .01) %>%
  filter(y_round %in% seq(from = 10, to = 40, by = 4)) %>%
  ggplot(aes(x*1000, h2o_conc*1000, color = mech)) +
```

```
  geom_line() +
  facet_wrap(~y_round) +
  labs(x = 'x coordinate (mm)', y = 'Concentration of H2O (mol/m^3)',
       title = 'H2O Concentration Profile at Select HABs for the 2 Mechanisms',
       subtitle = 'Between x values of -10 mm and 10 mm')+
  theme(text = element_text(size = 20)) +
  scale_color_discrete(labels = c('5-Step', 'USC2'))
```

Source code for soot production model

```
library(tidyverse)
library(scales)

# calculate the rate of soot production using formula from francesco
# formula is in notebook, date is 4/3/2020
# upload data in this folder

# take formula from Excel sheet Francesco sent
tib <- tib %>%
  mutate(k = 2e19*exp(-128/(1.99e-3*temperature)),
         soot_rate = k*c2h2_conc*1000*c6h6_conc*1000,
         y_round = y_round*1000)

# plot soot rate at different HABs
tib %>%
  filter(x < .01, x > -.01, y_round %in% seq(from = 10, to = 40, by = 4)) %>%
  mutate(y_round = paste('HAB =', paste(y_round, 'mm', sep = ' '), sep = ' ')) %>%
  ggplot(aes(x*1000, soot_rate)) +
  geom_point()+
  facet_wrap(y_round~.) +
```

```
  theme(legend.position="none") +
  labs(x = 'x Coordintate (mm)',
       y = 'Rate of Soot Production (mol/(m^3*s)',
       title = 'Profiles of Soot Production Rate') +
  theme(text = element_text(size = 20))

tib %>%
  filter(x < .0025, x > -.0025, y_round %in% seq(from = 10, to = 40, by = 4)) %>%
  mutate(y_round = paste('HAB =', paste(y_round, 'mm', sep = ' '), sep = ' ')) %>%
  ggplot(aes(x*1000, soot_rate, color = y_round)) +
  geom_point()+
  labs(x = 'x Coordintate (mm)',
       y = 'Rate of Soot Production (mol/(m^3*s)',
       title = 'Soot Production Rate on the Computational Area',
       color = 'HAB') +
  theme(text = element_text(size = 20))

# soot production rate with temp
tib %>%
  filter(x < .0025, x > -.0025, y_round %in% seq(from = 10, to = 40, by = 4)) %>%
  mutate(y_round = paste('HAB =', paste(y_round, 'mm', sep = ' '), sep = ' ')) %>%
  ggplot(aes(temperature, soot_rate, color = y_round)) +
  geom_point()+
  labs(x = 'Temperature (K)',
       y = 'Rate of Soot Production (mol/(m^3*s)',
       title = 'Soot Production Rate by Temperature',
       color = 'HAB') +
  theme(text = element_text(size = 20)) +
  xlim(c(750, 2200))
```

```
# recreate figure 5 from the effects of temperature on soot production paper
tib %>%
  filter(x < .01, x > -.01, y_round %in% seq(from = 10, to = 40, by = 4)) %>%
  mutate(y_round = paste('HAB =', paste(y_round, 'mm', sep = ' '), sep = ' ')) %>%
  ggplot(aes((1/temperature)*10^4, k)) +
  geom_point()+
  theme(legend.position="none") +
  labs(x = 'Inverse Temperature, 1/T^4 (K^-1)',
       y = 'k (m^3/(mol*s)') +
  theme(text = element_text(size = 20)) +
  xlim(c(4,9)) +
  scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                labels = trans_format("log10", math_format(10^.x))) +

# make the plot of concentrations of O2, C2H4, and H2O/CO2 in notes from 4/3/2020
dat <- tib %>%
  rename(h2o_conc = `concentration-h2o`,
         co2_conc = `concentration-co2`) %>%
  filter(x < .01, x > -.01) %>%
  select(y_round, c2h4_conc, o2_conc, h2o_conc, co2_conc, c6h6_conc, c2h2_conc) %>%
  rename(c2h4 = c2h4_conc,
         h2o = h2o_conc,
         o2 = o2_conc,
         co2 = co2_conc,
         c2h2 = c2h2_conc,
         c6h6 = c6h6_conc) %>%
  pivot_longer(-y_round, names_to = 'species_conc', values_to = 'conc')
```

```
dat %>%
  filter(species_conc %in% c('c2h4', 'o2', 'h2o', 'co2')) %>%
  ggplot(aes(y_round, conc, color = species_conc))+
  geom_smooth(se = F)+
  labs(x = 'HAB (mm)',
       y = 'Species Concentration (mol/m^3)',
       title = 'Change in Species Concentration',
       color = 'Species')+
  theme(text = element_text(size = 20))

dat %>%
  filter(species_conc %in% c('c2h4', 'o2', 'c2h2', 'c6h6')) %>%
  ggplot(aes(y_round, conc, color = species_conc))+
  geom_smooth(se = F)+
  labs(x = 'HAB (mm)',
       y = 'Species Concentration (mol/m^3)',
       title = 'Change in Species Concentration',
       color = 'Species')+
  theme(text = element_text(size = 20))
```